

**INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK ELEKTRONIKA**



**SKRIPSI**

**PERANCANGAN DAN PEMBUATAN SIMULASI  
PALANG PINTU KERETA API DAN “ TRAFFIC LIGHT “  
DI KOTA LAMA MALANG SECARA OTOMATIS  
BERBASIS MIKROKONTROLER AT89S8252**

*Disusun Oleh :*

**NAMA : JULI SURYA PURWANTO  
NIM : 02.17.017**

**MARET 2007**

---

# LEMBAR PERSETUJUAN



## PERANCANGAN DAN PEMBUATAN SIMULASI PALANG PINTU KERETA API DAN "TRAFFIC LIGHT" DI KOTA LAMA MALANG SECARA OTOMATIS BERBASIS MIKROKONTROLLER AT89S8252

### SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Teknik Elektronika Strata Satu (S-1)*

Disusun Oleh :

**JULI SURYA PURWANTO**  
NIM. 02.17.017

Diperiksa dan Disetujui

Dosen Pembimbing I

( Ir. Widodo Pudji Mulyanto, MT. )  
NIP.Y 1028700171

Dosen Pembimbing II

( I Komang Somawirata, ST., MT. )  
NIP.1030100361

Mengetahui



Ketua Jurusan Teknik Elektro S-1

( Ir. E. Yadi Limpraptono, MT )

NIP.Y. 1039500274

KONSENTRASI TEKNIK ELEKTRONIKA  
JURUSAN TEKNIK ELEKTRO S-1  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2007



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI T.ELEKTRONIKA

## BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : Juli Surya Purwanto  
Nim : 02.17.017  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : Perancangan dan Pembuatan Simulasi Palang Pintu Kereta  
Api dan "Traffic Light" di Kota Lama Malang Secara  
Otomatis Berbasis Mikrokontroller AT89S8252

Dipertahankan Dihadapan Tim Penguji Skripsi Jenjang Strata Satu (S-1) Pada :

Hari : Sabtu  
Tanggal : 17 Maret 2007  
Dengan Nilai : A (83,5) *84*



( Ir. Mochtar Asroni, MSME. )  
NIP.Y. 1018100036

Panitia Ujian Skripsi

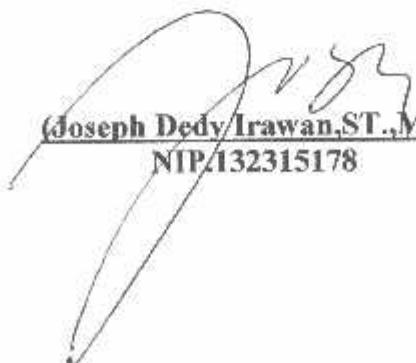
Sekretaris



( Ir.F.Yudi Limpraptono,MT. )  
NIP.1039500274

Anggota Penguji

Penguji I



( Joseph Dedy Irawan,ST.,MT. )  
NIP.132315178

Penguji II



( M. Ashar,ST.,MT. )

## ABSTRAKSI

### PERANCANGAN DAN PEMBUATAN SIMULASI PALANG PINTU KERETA API DAN “TRAFFIC LIGHT” DI KOTA LAMA MALANG SECARA OTOMATIS BERBASIS MIKROKONTROLLER AT89S8252

( Juli Surya Purwato, 02.17.017, Jurusan Teknik ElektroS-1/Elektronika )  
( Dosen Pembimbing I : Ir. Widodo Pudji Mulyanto, MT. )  
( Dosen Pembimbing II : I Komang Somawirata, ST. )

---

Kata Kunci : mikrokontroller AT89S8252, motor DC, sensor rel kereta api, *traffic light*

Saat ini kendali pintu lintasan kereta api masih dilakukan secara *manual* oleh seorang petugas jaga pintu lintasan kereta api. Apabila petugas jaga lalai dalam menjalankan tugasnya maka kecelakaan tidak dapat dihindarkan lagi. Dengan dibuatnya alat ini maka diharapkan kecelakaan akibat kelalaian petugas jaga dapat dihindarkan

Sistem ini menggunakan Mikrokontroller AT89S8252 sebagai basisnya. Alat yang dibuat meliputi perencanaan perangkat keras dan perangkat lunak. Perencanaan perangkat keras meliputi: rangkaian Sensor rel kereta api, rangkaian *driver* motor DC, rangkaian *traffic light*, rangkaian *driver Buzzer*, minimum sistem Mikrokontroller AT89S8252. Perencanaan perangkat lunak berupa *Flowchart* cara kerja sistem. Palang pintu lintasan kereta api akan menutup dan lampu *traffic light* menyala merah semua, kemudian apabila palang pintu lintasan kereta api membuka maka *traffic light* kembali normal ditandai dengan berbunyi *Buzzer* secara otomatis apabila mendapat sinyal dari sensor-sensor yang ada.

Dari hasil pengujian rangkaian sensor rel kereta api yang prinsip kerjanya sama dengan *switch*, maka dapat diketahui bahwa pada saat ada *penekanan* ( ada kereta ) saklar akan menutup dan tegangan pada P1.0 adalah 0,00 Volt, sebaliknya pada saat kondisi *tidak ada penekanan* ( tidak ada kereta ) maka saklar akan terbuka dan tegangan pada P1.0 adalah 5,52 Volt. Pada rangkaian *driver* motor DC besarnya arus yang mengalir pada saat kondisi relay aktif adalah 93,3 mA, sedangkan arus yang mengalir pada saat motor aktif adalah 133 mA. Sehingga dari hasil pengujian maka rangkaian *driver* motor tersebut dapat membuka dan menutup palang pintu lintasan kereta api dengan baik. Rangkaian *traffic light* juga dapat bekerja dengan baik. Ini dibuktikan dengan kecilnya prosentase *error*. *Error* pada rangkaian *traffic light* yaitu sebesar 1,32 %. Dari hasil pengujian seluruh rangkaian maka alat ini dapat bekerja sesuai dengan perencanaan.

## KATA PENGANTAR

Atas Berkat Rahmat Allah Yang Maha Kuasa, sehingga penulis dapat menyelesaikan laporan Skripsi dengan judul :

### **PERANCANGAN DAN PEMBUATAN SIMULASI PALANG PINTU KERETA API DAN “TRAFFIC LIGHT” DI KOTA LAMA MALANG SECARA OTOMATIS BERBASIS MIKROKONTROLLER AT89S8252**

Pembuatan Skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata-1 di Institut Teknologi Nasional Malang. Laporan Skripsi ini merupakan tanggung jawab tertulis atas ilmu pengetahuan yang didapat selama penyusun mengikuti kuliah.

Atas terselesaikannya Skripsi ini, penulis mengucapkan terima kasih kepada :

- Bapak Prof.Dr. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
- Bapak Ir. Mochtar Asroni, MSME selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
- Bapak Ir.F.Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro SI.
- Bapak Ir. Widodo Pudji Mulyanto, MT selaku Dosen Pembimbing I yang telah memberikan bimbingan, pengarahan, serta ilmu-ilmu yang sangat berharga sehingga Skripsi ini dapat terselesaikan.

- Bapak I Komang Somawirata, ST., MT. selaku Dosen Pembimbing II yang telah memberikan bimbingan, pengarahan, serta ilmu-ilmu yang sangat berharga sehingga Skripsi ini dapat terselesaikan.
- Bapak Ir. Yusuf Ismail Nahkoda, MT. selaku Sekretaris Jurusan Teknik Elektro S1.

Penulis mengharapkan kritik dan saran yang membangun untuk kesempurnaan skripsi ini.

Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan Skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga Skripsi ini dapat bermanfaat bagi kita semua.

Malang,     Maret 2007

Penulis

## DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	i
LEMBAR PERSETUJUAN .....	ii
ABSTRAKSI .....	iii
KATA PENGANTAR .....	iv
DAFTAR ISI .....	vi
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	xi
BAB I. PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	1
1.3. Tujuan .....	2
1.4. Batasan Masalah .....	2
1.5. Metodologi .....	2
1.6. Sistematika Penulisan .....	3
BAB II. LANDASAN TEORI .....	4
2.1. Mikrokontroller AT89S8252 .....	4
2.1.1. Perangkat keras Mikrokontroller AT89S8252 .....	5
2.1.2. SFR tambahan pada Atmel AT89S8252 .....	11
2.1.3. Data Memory (EEPROM) dan RAM .....	11

2.1.4. Programable Watchdog Timer (WDT).....	12
2.1.5. Timer 2.....	13
2.2. Motor DC .....	15
2.2.1. Cara Kerja Motor DC.....	16
2.2.2. Pengendalian Arah Putaran Motor DC .....	18
2.3. Relay.....	18
2.4. IC ULN 2003A.....	20
2.5. <i>Limit Swicth</i> .....	22
2.6. LED ( <i>Light Emitting Dioda</i> ) .....	22
2.7. Buzzer.....	23
 <b>BAB III. PERENCANAAN DAN PEMBUATAN ALAT</b> .....	 24
3.1. Perencanaan Perangkat Keras .....	24
3.1.1. Sensor Rel Kereta Api.....	27
3.1.2. Minimum Sistem AT89S8252 .....	29
3.1.3. Rangkaian <i>Traffic Light</i> .....	30
3.1.4. Rangkaian Driver Motor DC.....	32
3.1.5. Rangkaian Driver <i>Buzzer</i> .....	34
3.2. Perancangan Perangkat Lunak .....	34
3.2.1. Diagram Alir .....	35
 <b>BAB IV. PENGUJIAN ALAT</b> .....	 36
4.1. Pengujian Alat Setiap Blok .....	36
4.1.1. Pengujian Sensor Rel Kereta Api.....	36



4.1.2. Pengujian Driver Motor .....	38
4.1.3. Pengujian Rangkaian Buzzer .....	40
4.1.4. Pengujian Traffic Light .....	42
4.2. Pengujian Sistem Secara Keseluruhan .....	48
4.3. Spesifikasi alat.....	49
 <b>BAB V. PENUTUP</b> .....	 51
5.1. Kesimpulan.....	51
5.2. Saran.....	52

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

## DAFTAR GAMBAR

Gambar	Halaman
2.1 Blok Diagram Mikrokontroler AT89S8252 .....	6
2.2 Mikrokontroler AT89S8252.....	7
2.3 Garis-garis Medan Magnet di Sekitar Arus listrik pada Kawat Lurus ...	15
2.4 Kaidah tangan Kanan .....	16
2.5 Cara Kerja Motor DC.....	16
2.6 Sebuah Motor DC.....	17
2.7 Pengendalian Arah Putaran Motor DC .....	18
2.8 Cara kerja Relay .....	18
2.9 Relay SPST .....	19
2.10 Relay SPDT.....	19
2.11 Relay DPST.....	19
2.12 Relay DPDT .....	20
2.13 Pin-pin Koneksi dalam IC ULN 2003A .....	21
2.14 Rangkaian Darlington di dalam IC ULN 2003A .....	21
2.15 Simbol Limit Switch .....	22
2.16 Simbol Led .....	23
2.17 Bentuk Fisik <i>Buzzer</i> .....	23
3.1 Diagram Blok Sistem Pengendali Palang Pintu Lintasan Kereta Api dan traffic light.....	25
3.2 Tata Letak Sistem Pengendali Palang Pintu Lintasan Kereta Api dan traffic light.....	26

3.3 Celah Sambungan Rel Kereta Api .....	28
3.4 Potongan Rel Kereta Api Tampak dari Depan dan Tampak dari Samping.....	28
3.5 Sambungan Rel Kereta Api Sebelum Dimodifikasi.....	28
3.6 Sambungan Rel Kereta Api Setelah Dimodifikasi.....	29
3.7 Rangkaian Detektor.....	29
3.8 Rangkaian Minimum Sistem AT89S8252 .....	30
3.9 Rangkaian <i>Traffic Light</i> .....	31
3.10 Rangkaian <i>Driver Motor DC</i> .....	33
3.11 Rangkaian <i>Driver Buzzer</i> .....	34
4.1 Pengujian Detektor Rel Kereta Api.....	37
4.2 Pengujian Rangkaian <i>Driver Relay</i> .....	39
4.3 Pengujian Rangkaian Buzzer .....	41
4.4 Pengujian <i>Traffic Light</i> .....	43
4.5 Simulasi Alat.....	50

## DAFTAR TABEL

<b>Tabel</b>	<b>Halaman</b>
2.1 Konfigurasi Port 3 Atmel 89S8252.....	9
2.2 Fungsi khusus pada Port 1 Atmel 89S8252 .....	10
2.3 Pemilihan periode Waktu WDT.....	13
2.4 Mode Operasi Timer 2 .....	14
3.1 Pola Pengaturan Nyala Lampu Traffic Light .....	31
4.1 Hasil Pengujian Rangkaian Tombol Start.....	37
4.2 Hasil Pengukuran Arus Pada Motor.....	39
4.3 Hasil Pengukuran Arus Pada Relay .....	40
4.4 Hasil Pengukuran Arus Pada Rangkaian Buzzer .....	41
4.5 Pola Pengaturan Nyala Lampu Traffic Light .....	45
4.6 Hasil Pengujian Traffic Light.....	47

## DAFTAR LAMPIRAN

### Lampiran

1. Lampiran 1
  - Formulir –Formulir Skripsi
2. Lampiran 2
  - Gambar Rangkaian
3. Lampiran 2
  - Listing Program
4. Lampiran 3
  - Data Sheet Mikrokontroler AT89S8252
5. Lampiran 4
  - Data Sheet IC ULN 2003



# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam hal pengaturan lalu lintas, teknologi elektronika belum terlalu banyak digunakan. Hal ini dapat kita lihat pada lintasan kereta api khususnya di daerah perempatan jalan raya. Pada perempatan jalan raya yang dilintasi kereta api sering kali kita lihat adanya kemacetan-kemacetan lalu lintas yang disebabkan kurang teraturnya penataan dari lintasan kereta api tersebut. Selain itu pada lintasan-lintasan kereta api juga sering kali terjadi kecelakaan yang diakibatkan kurang efisiennya penggunaan palang pintu kereta api.

Berdasarkan realita tersebut dan untuk mengatasi kondisi di atas, dalam tugas akhir ini akan dirancang palang pintu kereta api secara otomatis dengan traffic light dengan berbasis mikrokontroller AT89S8252. Dalam laporan akhir ini digunakan sambungan rel kereta api sebagai sensor untuk mendeteksi adanya kereta api yang lewat.

### **1.2 Rumusan Masalah**

Rumusan masalah tersebut adalah bagaimana cara mengetahui kedatangan kereta api dengan lebih mudah dan aman serta membuat palang pintu kereta api pada perempatan jalan raya yang dilengkapi traffic light dengan menggunakan mikrokontroller AT89S8252.

### 1.3 Tujuan

Tujuan penulisan skripsi ini adalah merancang dan membuat palang pintu kereta api pada perempatan jalan raya yang dilengkapi *traffic light* berbasis mikrokontroller AT89S8252. Pembuatan alat tersebut diharapkan dapat mengurangi angka kemacetan dan kecelakaan lalu lintas khususnya pada perempatan jalan raya yang dilintasi kereta api.

### 1.4 Batasan Masalah

Agar permasalahan tidak terlalu luas serta sistematis dan terarah, maka penulis membatasi masalah hanya pada hal-hal berikut :

1. Pengolahan data utama adalah mikrokontroller AT89S8252.
2. Alat yang dirancang dan dibuat dalam bentuk simulasi.
3. Tidak membahas masalah perpindahan jalur rel kereta api.
4. Untuk mendeteksi kedatangan kereta api digunakan sambungan rel kereta api yang telah dimodifikasi.

### 1.5 Metodologi

Metode yang digunakan dalam perencanaan dan pembuatan laporan tugas akhir ini adalah :

#### 1. *Study Literature*

Dengan mempelajari teori serta aplikasi dari mikrokontroller AT89S8252 sebagai pengolah data.



## 2. *Field Research*

Melakukan percobaan dan membandingkan data hasil percobaan dengan kondisi sebenarnya di lapangan yang merupakan data primer sebagai pembuktian kebenaran dari data sekunder hasil studi literatur.

3. Merancang dan membuat perangkat keras dan perangkat lunaknya.
4. Uji coba terhadap alat yang telah dibuat.
5. Menyusun laporan skripsi.

## 1.6 **Sistematika**

Sistematika dari penyusunan laporan skripsi ini adalah :

### BAB I     PENDAHULUAN

Berisi latar belakang belakang, rumusan masalah, tujuan, pembatasan masalah, metodologi penulisan, serta sistematika penyusunan dan pembuatan alat.

### BAB II    TEORI PENUNJANG

Berisi tentang teori – teori dasar yang memiliki relevansi sebagai dasar perencanaan dan pembuatan alat.

### BAB III   PERENCANAAN DAN PEMBUATAN

Berisi tentang perencanaan hardware dan software.

### BAB IV   PENGUJIAN ALAT

Berisi tentang data hasil pengujian peralatan yang telah dibuat secara keseluruhan.

### BAB V    PENUTUP

Berisi kesimpulan dan saran dari skripsi ini



## BAB II

### TEORI DASAR

#### 2.1 Mikrokontroller AT89S8252

Mikrokontroller memiliki beberapa perbedaan yang sangat mendasar apabila dilihat dari segi fungsi serta penggunaannya. Selain terdapat CPU pada Mikrokontroller juga terdapat memori I/O yang merupakan kelengkapan suatu sistem minimum dari mikrokomputer. Karena itu dapat dikatakan bahwa mikrokontroller merupakan mikrokomputer dalam keping tunggal (*single chip microcomputer*) yang dapat berdiri sendiri.

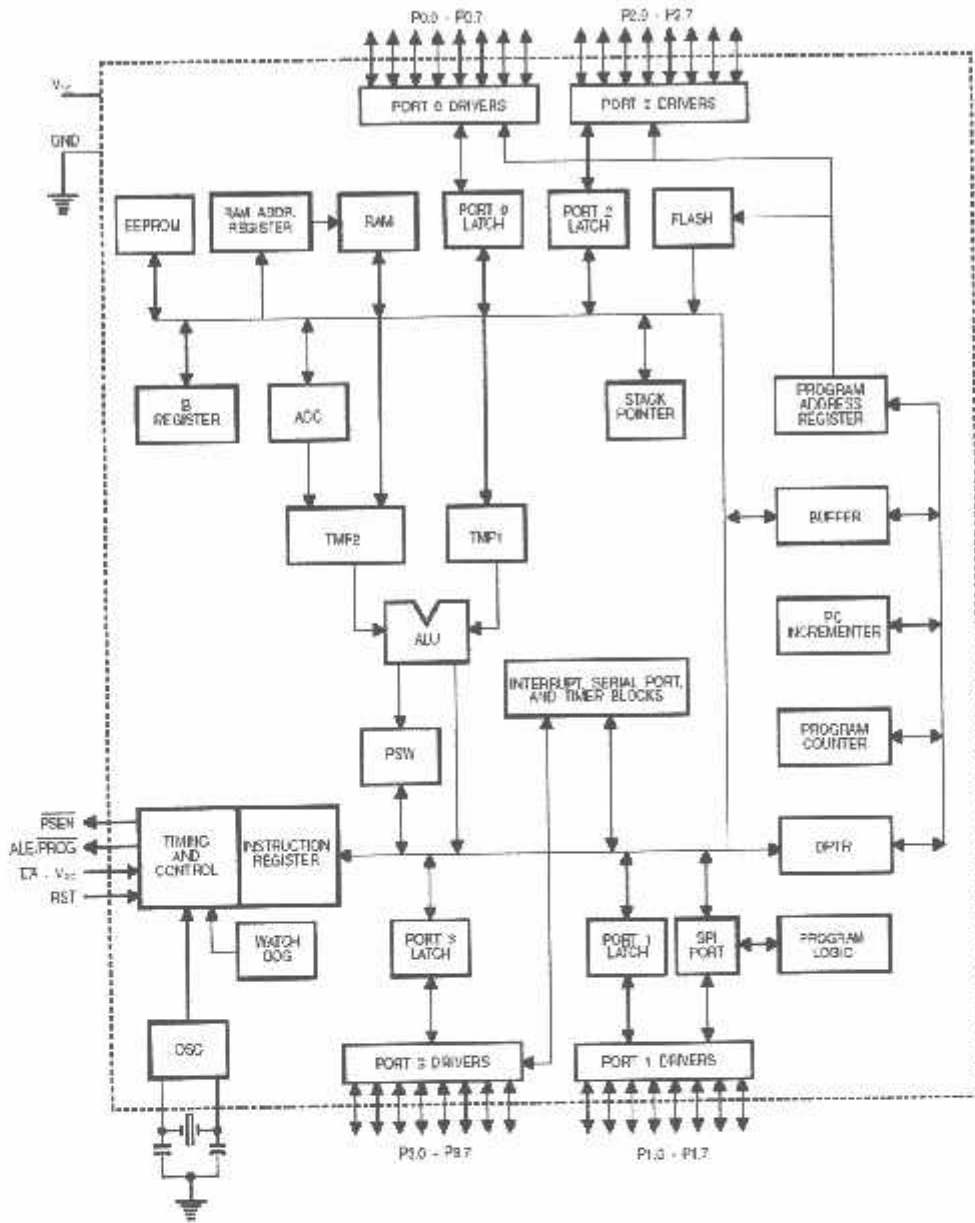
Mikrokontroller AT89S8252 adalah mikrokontroller ATMEL kompetibel penuh dengan mikrokontroller keluarga MCS-51, dengan supply daya rendah, memiliki *performance* yang tinggi, dan merupakan mikrokontroller 8 k byte Downloadable Flash Memori yang dilengkapi 2 Kbyte EEPROM ( *Electrical Erasable and programmable read Only Memory*) dan 256 Byte RAM internal.

Dalam sistem mikrokontroller terdapat dua hal yang mendasar, yaitu: perangkat keras dan perangkat lunak yang keduanya saling terkait dan mendukung.

### 2.1.1 Perangkat keras Mikrokontroller AT89S8252

Mikrokontroller AT89S8252 merupakan pengembangan dari mikrokontroller standard MCS-51, dengan banyak kelebihan yang ditawarkan antara lain :

- Kompatibel dengan Mikrokontroller MCS – 51
- 8K byte Downloadable Flash memori
- 2K byte EEPROM
- 3 level program memori lock
- 256 byte RAM internal
- 32 I/O yang dapat dipakai semua
- 3 buah timer / counter 16 bit
- Programmable watchdog timer
- Dual data pointer
- Frekuensi kerja 0 sampai 24 MHZ
- Tegangan operasi 2,7 – 6 volt

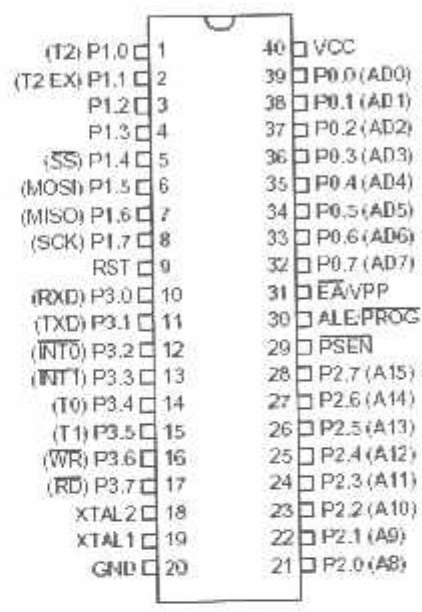


**Gambar 2.1 Blok Diagram Mikrokontroller AT89S8252**

Dipakainya downloadable flash memori memungkinkan mikrokontroller ini bekerja sendiri tanpa diperlukan tambahan chip lainnya. Dan flash memori dapat diprogram hingga seribu kali.

*Timer / counter* juga bertambah satu dari standar 2 buah pada MCS – 51. Selain itu frekuensi kerja yang lebar dan rancangan statik sangat membantu untuk

proses *debugging*. Adanya beberapa fitur tambahan itu, mengakibatkan bertambahnya SFR (*Special Function Register*). Gambar berikut adalah gambar mikrokontroler AT 89S8252. Tata letak pin – pin ini masih mengacu pada mikrokontroler MCS – 51 sehingga AT 89S8252 dapat menggantikan mikrokontroler <sup>1</sup>MCS – 51.



**Gambar 2.2 Mikrokontroler AT 89S8252**

Keterangan pin :

- Pin 40 (VCC)  
Merupakan pin catu daya dengan tegangan sebesar +5 V (DC )
- Pin 20 ( GND )  
Merupakan pin GROUND yang nanti terhubung dengan grounding rangkaian.

- Pin 32 – 39 ( Port 0 )

PORT 0 mempunyai fungsi sebagai port alamat dan data , jika mikrokontroler sedang mengakses alamat, P0 akan aktif sebagai pembawa alamat 8 bit yang bawah ( A0 – A8 ). Ketika mengakses data (bisa input atau output) port ini sebagai jalur data ( D7 – D0 ).

- Pin 21 – 18 ( Port 2 )

Port 2 berfungsi sebagai pembawa alamat 8 bit atas (A8 – A15). Berbeda dengan port 0, port ini tidak bersifat sebagai jalur data hanya sebagai pembawa alamat. Dengan demikian jelas bahwa untuk alamat AT89S8252 menyediakan 16 bit sedangkan untuk jalur data hanya 8 bit.

- Pin 10 – 17 ( port 3 )

Port 3 ini mempunyai fungsi yang berlainan dari setiap pin-pinnya, seperti yang ditunjukkan di bawah ini:

-P3.7: Kaki *read* yang aktif manakala sedang melakukan eksekusi yang sifatnya membaca data.

-P 3.6 : Kaki *write* yang aktif saat melakukan eksekusi yang sifatnya menulis data ke suatu alamat

-P 3.5 :Merupakan pin yang berhubungan dengan timer register timer 1 (T1).

-P 3.4 : Merupakan pin yang berhubungan dengan timer register 0 (T0).

-P 3.3 : Berhubungan dengan control interrupt (INT1)

-P 3.2 : Berhubungan dengan control interrupt (INT0)

-P 3.1 : Berhubungan dengan port serial (TXD)

-P 3.0 : Berhubungan dengan port serial (RXD)

Untuk lebih jelasnya lihat daftar tabel di bawah ini :

**Tabel 2.1 Konfigurasi Port 3 Atmel 89S8252**

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

- Pin 9 (RST (*Reset*))

Pin reset ini aktif tinggi (1), jika pin ini aktif tinggi minimal dua kali siklus mesin bekerja maka akan mereset peralatan.

- Pin 30 (ALE/PROG (*Address Latch Enable/Program*))

Pin ALE ini aktif tinggi dengan mengeluarkan pulsa output untuk mengunci/menahan 1 byte alamat rendah selama mengakses ke alamat memori eksternal. ALE dapat mengendalikan 8 beban TTL dan juga merupakan input program yang aktif rendah selama pemrograman *Flash Eprom*. Pada operasi normal, ALE dikeluarkan pada suatu kecepatan yang konstan yaitu 1/6 dari frekuensi osilator, dan juga dapat dipergunakan untuk pewaktu atau timing eksternal atau untuk pemberian clock.



- Pin 29 (PSEN (*Program Strobe Enable*))

Pin ini aktif rendah, yang merupakan pulsa pengaktif untuk pembacaan ke program memori eksternal.

- Pin 19 (XTAL - !)

Sebagai pin input ke penguat osilator pembalik dan input rangkaian clock internal untuk operasi system.

- Pin 18 ((XTAL - 2)

Pin output dari penguat osilator.

- Pin 31 (EA/VPP (*Externall Access/Program Supplay Voltage*))

Pin ini harus ditahan dalam kondisi rendah secara eksternal atau dihubungkan ke-ground agar AT 89S8252 dapat mengakses kode mesin dari memori eksternal. Jika menggunakan internal program memori, pin ini harus diberi logika tinggi (1).

Pada port 1 mikrokontroler AT 89S8252 terdapat beberapa fungsi khusus yang tidak terdapat pada mikrokontoler MCS-51. Fungsi khusus tersebut dijelaskan pada tabel berikut :

**Tabel 2.2 Fungsi khusus pada Port 1 Atmel 89S8252**

Port PIN	Fungsi Khusus
P 1.0	T2 (masukan luar untuk timer / counter 2)
P 1.1	T2 EX (timer / counter capture / reload trigger dan control arah)
P 1.2	-
P 1.3	-

P 1.4	SS ( <i>slave port select input</i> )
P 1.5	MOSI (master data output, slave dan input untuk kanal SPI)
P 1.6	MISO (master data input, slave data output untuk kanal SPI)
P 1.7	SCK (master clock output, slave clock input untuk kanal SPI)

### 2.1.2 SFR tambahan pada Atmel AT89S8252

Selain memiliki SFR (*Special Function Register*) seperti halnya pada MCS-51, mikrokontroler Atmel AT89S8252 memiliki tambahan SFR. Hal ini tak lain karena adanya fitur tambahan pada mikrokontroler Atmel AT89S8252.

SFR tambahan ini meliputi: T2CON (*Timer 2 Register* dengan alamat 0C8H), T2MOD (*Timer 2 Mode* dengan alamat 0C9H), WMCON (*Watchdog and Memory Control Register* dengan alamat 96H), SPCR(*SPI Control Register* dengan alamat D5H), SPSR (*SPI Status Register* dengan alamat AAH), SPDR(*SPI Data Register* dengan alamat 86H).

### 2.1.3 Data Memory (EEPROM) dan RAM

Berbeda dengan mikrokontroler standar MCS-51, mikrokontroler Atmel AT89S8252 juga dilengkapi dengan data memori yang berupa EEPROM (*Electrically Erasable Programmable Read Only Memory*). EEPROM yang ditanamkan ini besarnya 2 kilo byte (2K) dan dipakai untuk penyimpanan data.

EEPROM ini diakses dengan mengeset bit EEMEN pada register WMCON pada alamat 96H. Alamat EEPROM ini adalah 000H sampai 7FFH. Instruksi movx digunakan untuk mengakses EEPROM internal ini. Namun jika ingin mengakses data memori luar (diluar mikrokontroler Atmel AT89S8252) dengan menggunakan instruksi movx ini maka bit EEMEN harus dibuat "0".

Bit EEMWE pada register WMCON harus diset ke 1 sebelum sembarang lokasi pada EEPROM dapat ditulisi. Program pengguna harus mereset bit EEMWE ke "0" jika proses penulisan ke EEPROM tidak diperlukan lagi. Proses penulisan pada EEPROM dapat dilihat dengan membaca bit RDY/BSY pada SFR WMCON. Jika bit ini berlogika rendah maka penulisan EEPROM sedang berlangsung, jika bit ini berlogika tinggi maka penulisan sudah selesai dan penulisan lain dapat dimulai lagi. Sedangkan RAM yang ada pada mikrokontroler Atmel AT89S8252 adalah berkapasitas 256 byte. Penjelasan mengenai RAM ini adalah sama dengan RAM yang ada pada mikrokontroler standart MCS-51.

#### **2.1.4 Programable Watchdog Timer (WDT)**

Mikrokontroler Atmel AT89S8252 juga dilengkapi oleh *watchdog Timer*. *Watchdog Timer* ini menggunakan detak tersendiri. Untuk mengatur rentang waktu (periode) pada WDT ini terdapat bit prescaler yang dapat mengatur rentang waktu yang dibutuhkan.

Bit prascaler ini adalah bit PS0, PS1 dan PS2 pada register WMCON. Periode waktu pada WDT ini berkisar dari 16 mili detik sampai 2048 mili detik. Karena bit prescalernya ada 3, maka akan ada 8 buah kemungkinan yaitu:

**Tabel 2.3 Pemilihan Perioda Waktu WDT**

PS2	PS1	PS0	Perioda
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

WDT dinonaktifkan oleh Power on Reset (POR) dan selama Power Down. WDT diaktifkan dengan mensetting bit WDTEN pada SFR WMCON (alamat 96H). Jika perhitungan waktu WDT telah selesai tanpa ada reset atau dilumpuhkan, maka suatu pulsa reset internal akan dihasilkan untuk mereset CPU.

### 2.1.5 Timer 2

Pada mikrokontroler Atmel AT89S8252 terdapat tambahan *Timer 2*. *Timer* yang lain adalah *timer 0* dan *timer 1*. Hal yang perlu diperhatikan adalah *Timer/Counter* dapat digunakan sebagai generator *baudrate* untuk serial port. Pada standart MS-51 biasanya yang digunakan adalah *timer 1* sebagai penghasil *baudrate*. Pada mikrokontroler Atmel AT89S8252 selain menggunakan *timer 1* sebagai *baudrate* (untuk menjaga kompatibilitas dengan MCS-51) juga dapat menggunakan *Timer 2* sebagai penghasil *baudrate* untuk serial port. *Timer 2* ini merupakan *Timer/Counter* yang berukuran 16 bit yang dapat beroperasi sebagai timer atau dapat beroperasi sebagai penghitung kejadian dengan detak dari luar. Pengaturan fungsi ini dilakukan dengan mengatur bit C/T2 pada SFR T2CON.

Terlihat bahwa jika bit ini tinggi maka akan terpilih fungsi *counter*, tapi jika bit ini rendah maka akan terpilih fungsi *Timer 2*.

*Timer 2* ini memiliki 3 mode operasi yaitu: *capture*, *auto reload (up dan down counting)* dan *baud rate generator*. Untuk memilih mode ini dilakukan dengan mengatur bit pada SFR *T2CON*.

*Timer 2* ini terdiri dari 2 buah *timer 8 bit register* yaitu *TH2* dan *TL2* dinaikkan tiap siklus mesin. Karena siklus mesin terdiri dari 1 *periode* osilasi, maka *count rate* menjadi 1/12 dari frekuensi osilator.

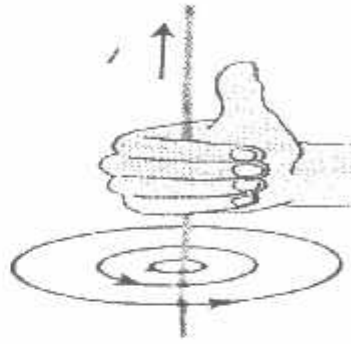
Pada fungsi *counter*, *register* dinaikkan berdasarkan tanggapan adanya transisi tinggi ke rendah pada pena yang bersesuaian (dalam hal ini pin *T2* atau *Pi.0*). pada fungsi ini, masukan luar akan disampling selama *S5P2* dari tiap siklus mesin. Jika hasil sampling menunjukkan logika tinggi pada selama satu siklus dan logika rendah pada siklus selanjutnya maka akan terdeteksi transisi tinggi ke rendah dan akibatnya perhitungan akan dinaikkan. Nilai perhitungan yang baru akan muncul pada register selama *S3P1* dari siklus setelah transisi tinggi ke rendah terdeteksi.

**Tabel 2.4 Mode Operasi Timer 2**

RCLK+TCLK	CP/RL2	TR2	MODE
0	0	1	16 bit auto reload
0	1	1	16 bit capture
1	X	1	Baud rate generator
X	X	0	Off

## 2.2 Motor DC

Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kanan. Ibu jari tangan menunjukkan arah aliran arus listrik sedangkan jari-jari yang lain menunjukkan arah medan magnet yang timbul, seperti yang ditunjukkan oleh gambar berikut ini.



**Gambar 2.3 Garis-Garis Medan Magnet di Sekitar Arus Listrik pada Kawat Lurus**

Kaidah tangan kanan untuk motor menunjukkan arah arus yang mengalir di dalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah gaya putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan :

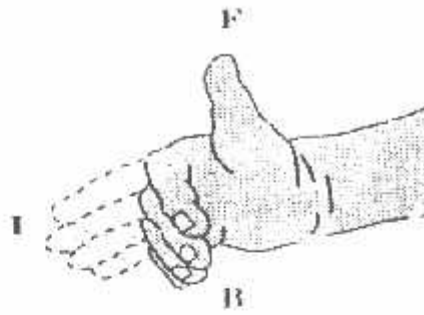
$$F = B.I.L \sin \theta \quad (\text{Newton})$$

Dimana :  $B$  = kerapatan fluks magnet (weber)

$L$  = panjang konduktor (meter)

$I$  = arus listrik (ampere)

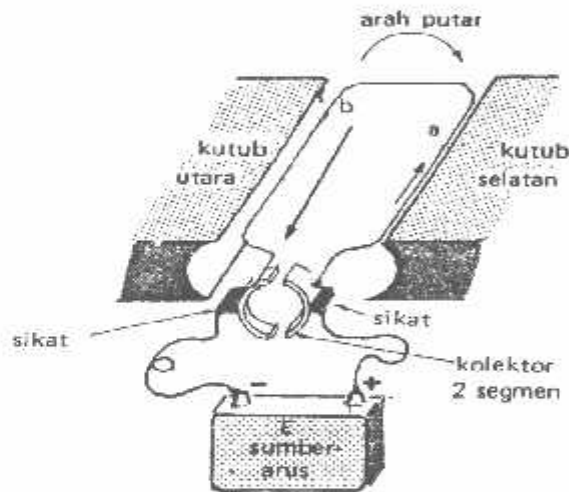
$\sin \theta$  = sudut antara antara arus dengan garis-garis medan.



**Gambar 2.4 Kaidah Tangan Kanan**

### 2.2.1 Cara Kerja Motor DC

Cara kerja motor DC dapat dilihat pada gambar di bawah ini:



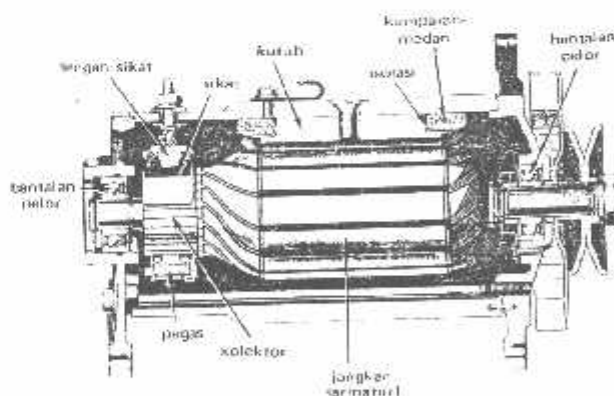
**Gambar 2.5 Cara Kerja Motor DC**

Ada satu lilit kawat a – b berada di dalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (*armour*).

Pada jangkar dimasukkan arus yang berasal dari sumber (baterai) E. koneksi baterai dengan jangkar melalui sikat-sikat. Sikat-sikat ini terpasang pada

sebuah cincin yang terbelah dua, yang disebut kolektor. Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatinya.

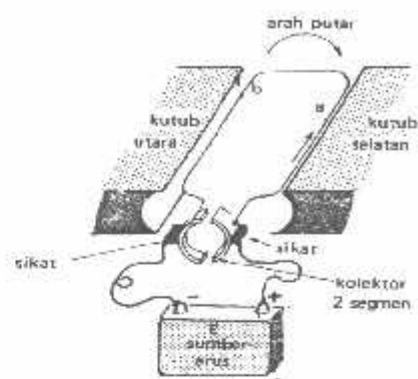
Pada kawat yang berada di kanan arus mengalir dari depan ke belakang . dalam kawat yang di kiri, arus mengalir dari belakang ke depan . kawat a dan b secara berganti-ganti berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu membolak balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertikal. Di sini kolektor berfungsi bagaikan penyearah mekanik. Fluk magnet yang ditimbulkan magnet permanen disebut medan magnetnya motor. Dalam gambar, arah fluk magnetik adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah . Gaya-gaya yang bekerja sama kuatnya, jadi ada kopel yang bekerja pada kawat sehingga lilitan pun dapat berputar. Setelah berputar  $90^0$  arah arus berbalik, pada saat itu penghantar a dan penghantar b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.



**Gambar 2.6 Sebuah Motor DC**



2.2.2 Pengendalian Arah Putaran Motor DC

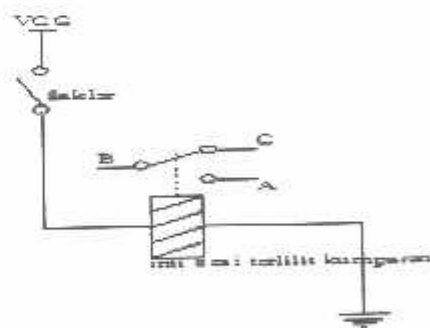


Gambar 2.7 Pengendalian Arah Putaran Motor DC

Berdasarkan gambar di atas, agar arah putaran motor DC berubah, polaritas tegangan pada baterai harus dibalik.

2.3 Relay

Relay adalah komponen elektronika yang terdiri dari sebuah lilitan kawat (kumparan/koil) yang terlilit pada sebuah besi lunak. Jika kumparan dialiri arus listrik maka inti besi akan menjadi magnet dan menarik pegas sehingga kotak AB terhubung dan BC terputus.

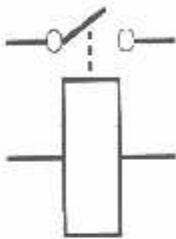


Gambar 2.8 Cara Kerja Relay

Relay merupakan suatu alat untuk menghubungkan atau memerlukan kontak antara komponen yang satu dengan yang lain. Dalam memutus atau menghubungkan kontak digerakkan oleh fluksi yang ditimbulkan dari adanya medan magnet listrik yang dihasilkan oleh kumparan yang melilit pada besi lunak.

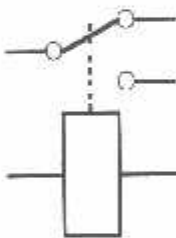
Ada beberapa macam relay, antara lain:

- SPST (*Single Pin Single Terminal*)



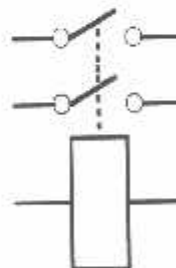
Gambar 2.9 Relay SPST

- SPDT (*Single Pin Dual Terminal*)



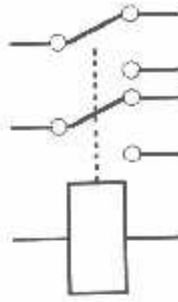
Gambar 2.10 Relay SPDT

- DPST (*Dual Pin Single Terminal*)



Gambar 2.11 Relay DPST

- DPDT (*Dual Pin Dual Terminal*)

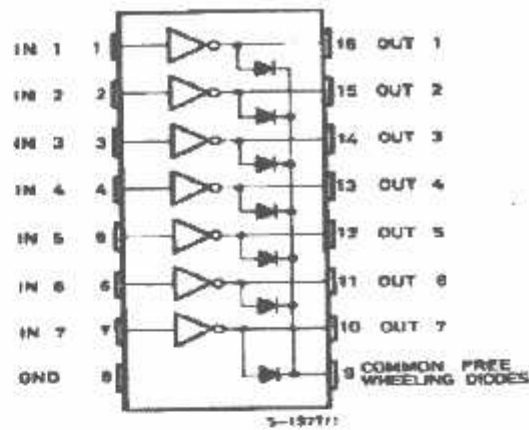


**Gambar 2.12 Relay DPDT**

## 2.4 IC ULN 2003A

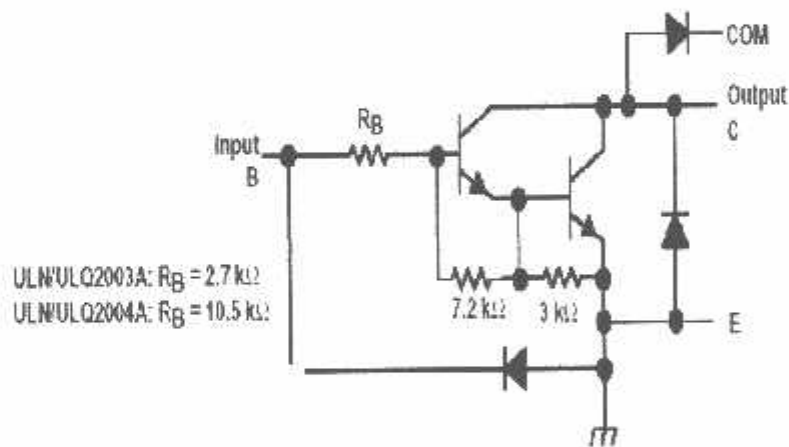
Pada IC ULN 2003A terdapat rangkaian penguat Darlington. Untuk satu IC ULN2003A terdapat 7 pasang rangkaian Darlington NPN yang tersusun dalam rangkaian *common catoda*. Rangkaian Darlington ini digunakan sebagai saklar. Pada masing-masing rangkaian Darlington arus kolektornya sebesar 500mA. Rangkaian Darlington yang ada di dalam IC ULN 2003A dapat diparalel untuk kebutuhan arus yang besar. Karenanya IC ini dapat diaplikasikan untuk *driver relay*, *driver lampu*, *driver display* dan *logic buffer*. Pada skripsi ini rangkaian Darlington yang ada pada IC ULN 2003A digunakan sebagai *driver relay* yang digunakan untuk mengendalikan putaran motor DC.

Pin-pin koneksi yang ada dalam IC ULN 20003A dapat dilihat pada gambar berikut ini:



**Gambar 2.13 Pin-Pin Koneksi dalam IC ULN 2003A**

Gambar untuk setiap rangkaian Darlington pada IC ULN 2003A dapat dilihat pada gambar berikut ini:



**Gambar 2.14**

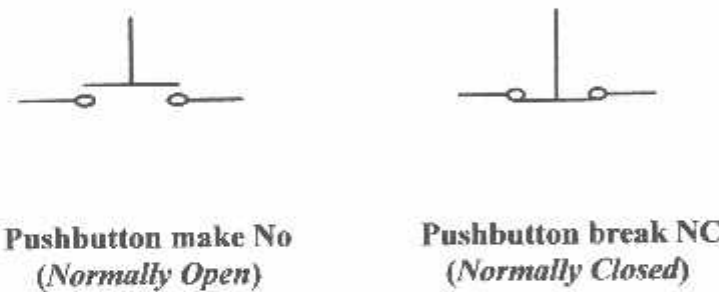
**Rangkaian Darlington di dalam IC ULN 2003A**

2.5 Limit Swicth

Limit swicth merupakan sebuah saklar yang bekerja karena ada suatu sentuhan atau gesekan. Limit swicth mempunyai beberapa bagian antara pengungkit dan roda penjulang yang merupakan bagian mekanik yang jika tersentuh oleh suatu benda atau mekanisasi lain, maka limit swicth akan on dan menggerakkan lengan pengungkit dalam suatu kontak.

Ada beberapa tipe limit switch yaitu limit swicth yang merupakan kontak NC (normally Closed) dan NO (Normally Open). Limit yang merupakan kontak NO berfungsi sebagai penghubung sedangkan yang kontak NC berfungsi sebagai pemutus.

Simbol dari limit swicth atau push button dari yang NC adalah sbb :



Gambar 2.15 Simbol Limit Switch

2.6 LED ( Light Emitting Dioda)

LED adalah suatu jenis dioda yang apabila diberi tegangan maju , arus majunya akan membangkitkan cahaya pada pertemuan PN-nya. Dioda-dioda yang digunakan terbuat dari bahan Galium (Ga), Arsen (As), dan Fosfor (P) atau disingkat GaAsP.

Tegangan maju antara anoda-katoda berkisar antara 1,5 – 2V, sedang arus majunya antara 5 – 20mA.



**Gambar 2.16 Simbol LED**

## **2.7 Buzzer**

*Buzzer* akan aktif apabila sensor aktif dan akan mengeluarkan isyarat berupa bunyi sebagai tanda apabila ada kereta api yang melintas. Gambar fisik dari *buzzer* adalah sebagai berikut :



**Gambar 2.17 Bentuk fisik *Buzzer***



### BAB III

#### PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas mengenai peralatan yang direncanakan dan akan direalisasikan sebagaimana fungsinya. Adapun perencanaan dan pembuatan alat meliputi : perencanaan dan pembuatan perangkat keras serta perencanaan dan pembuatan perangkat lunak secara garis besarnya.

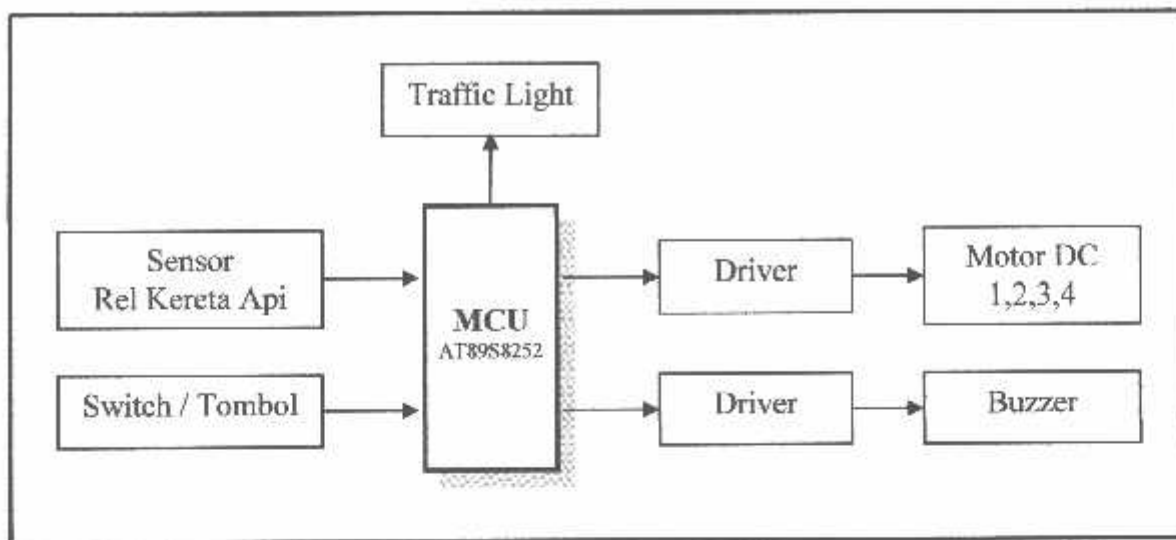
##### 3.1 Perencanaan Perangkat Keras

Pembuatan pengontrol elektronik sistem pengendali palang pintu lintasan kereta api ini hanya dalam bentuk simulasi, dengan perencanaan dan prinsip kerja yang sesuai dengan peralatan yang sebenarnya.

Otomatisasi yang diterapkan dalam perencanaan alat ini yaitu mengubah sistem atau prosedur lama yang telah ada, tidak menggunakan jalur komunikasi stasiun-pos lintasan dan tanpa keberadaan penjaga lintasan.

Komponen utama perangkat ini adalah Mikrokontroler AT89S8252 yang berfungsi sebagai unit pengendali otomatisasi pada palang lintasan kereta api. Berikut adalah diagram blok sistem pengendali palang pintu lintasan kereta api dan *traffic light*.



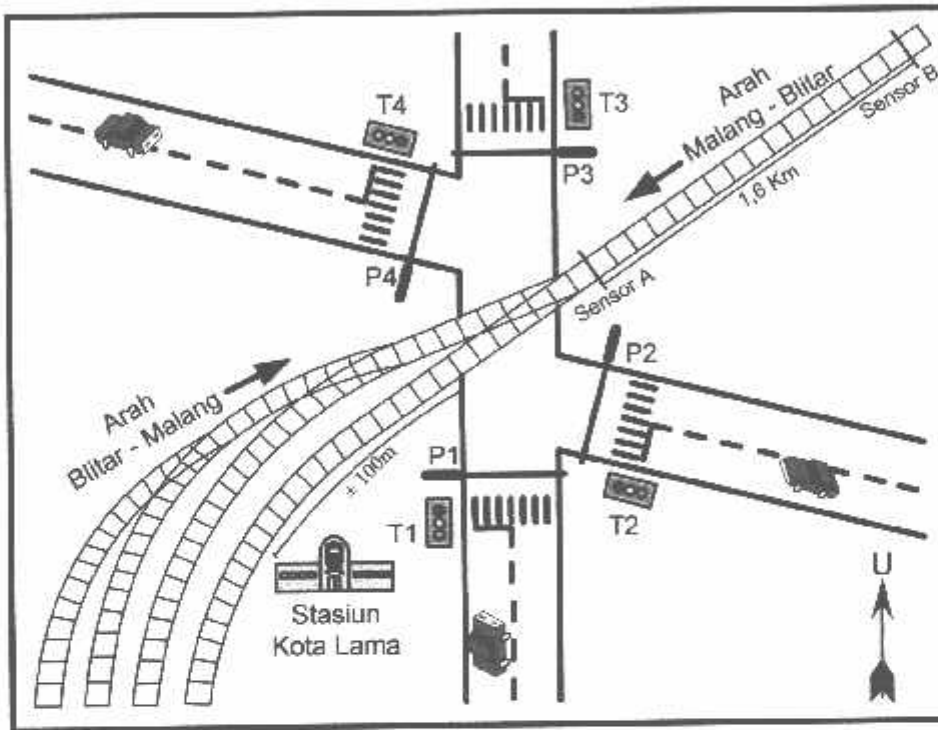


**Gambar 3.1 Diagram Blok Sistem Pengendali Palang Pintu Lintasan Kereta Api dan Traffic light**

Adapun cara kerja dari diagram blok di atas adalah sebagai berikut :

1. Sensor yang digunakan untuk mendeteksi kedatangan kereta api menggunakan celah sambungan rel kereta api yang telah dimodifikasi sehingga sensor ini bekerja seperti *switch*.
2. *Switch* / tombol berfungsi sebagai pengganti sinyal yang berasal dari stasiun kota lama apabila ada kereta api yang akan melintas menuju ke arah stasiun kota baru.
3. Mikrokontroller berfungsi sebagai pusat pengolahan data yang berasal dari sensor dan *switch*.
4. *Traffic light* untuk mengatur jalur lalu lintas yang ada pada perempatan jalan. Kerja dari *traffic light* ini diatur oleh mikrokontroller berdasarkan input dari sensor.

5. *Driver* digunakan untuk mengendalikan motor DC. Motor DC ini berfungsi untuk membuka dan menutup palang pintu lintasan kereta api.
6. *Driver Buzzer* digunakan untuk mengaktifkan buzzer. Buzzer akan aktif bila ada kereta api yang sedang melintas.



**Gambar 3.2. Tata Letak Sistem Pengendalian Palang Pintu Lintasan Kereta Api dan Traffic Light**

Keterangan :

1. Sensor A dan B, sebagai indikator kedatangan kereta api dengan menggunakan celah pembatas sambungan rel kereta api yang telah dimodifikasi sehingga ujung-ujungnya tidak terhubung. Prinsip kerja sensor ini sama dengan *switch*, sehingga sensor ini aktif jika tersentuh atau terhubung oleh roda kereta api yang terbuat dari besi.

Untuk jarak sensor A dan B ditentukan dengan memperhitungkan kecepatan maksimal kereta api dan waktu yang dibutuhkan untuk menutup palang pintu lintasan kereta api. Dengan memperkirakan kecepatan kereta maksimal 100 km/jam dan waktu yang dibutuhkan untuk menutup palang pintu lintasan kereta api kurang lebih 60 detik, maka dapat diketahui jarak sensor A dan B adalah :

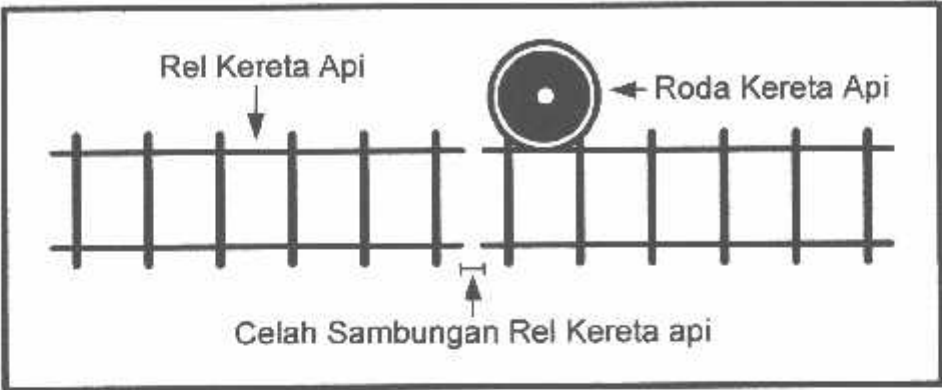
$$\begin{aligned} S &= v \cdot t \\ &= 100 \text{ km/jam} \times (60/3600) \text{ jam} \\ &= 1,66 \text{ km} \end{aligned}$$

2. T1,T2,T3,T4 adalah *Traffic light* yang berfungsi sebagai pengatur jalur lalulintas yang letaknya berada pada perempatan jalan raya. *Traffic light* ini bekerja secara normal, namun apabila ada kereta api yang akan melintas maka semua *traffic light* akan menyala merah . Kemudian apabila kereta api telah lewat maka *traffic light* akan kembali normal dengan mengambil data terakhir sebelum semua *traffic light* menyala merah.
3. P1,P2,P3,P4 adalah motor penggerak palang pintu lintasan kereta api yang aktif apabila ada kereta api yang akan melintas.

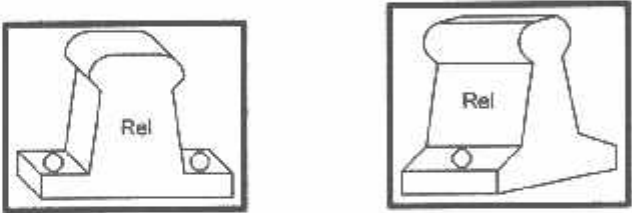
#### 3.1.1. Sensor Rel Kereta Api

Sensor ini memanfaatkan sambungan rel kereta api yang telah dimodifikasi sehingga ujung-ujung dari rel kereta api tidak terhubung dengan jarak 2 Cm. Sensor ini aktif apabila ada kereta api yang melintas sehingga kedua lajur rel kereta api akan terhubung akibat terkena roda kereta api yang terbuat dari

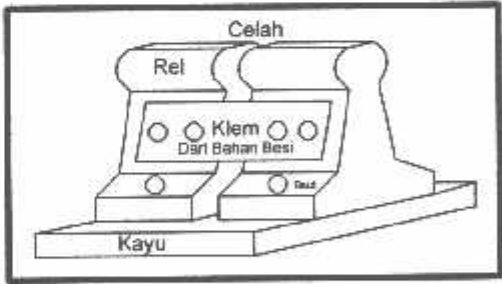
besi, sensor rel kereta api ini bekerja sesuai dengan prinsip kerja dari *switch*. Berikut ini gambar sambungan rel kereta api dan rangkaian sensor.



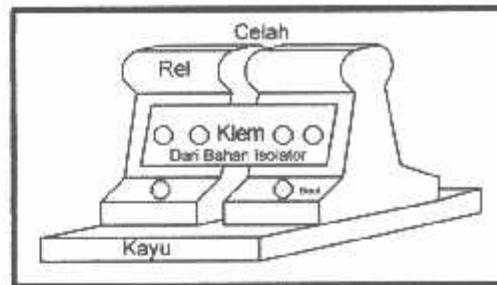
Gambar 3.3 Celah Sambungan Rel Kereta Api



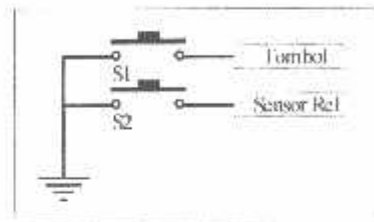
Gambar 3.4 Potongan Rel Kereta Api Tampak dari Depan dan Tampak dari Samping



Gambar 3.5 Sambungan Rel Kereta Api Sebelum Dimodifikasi



**Gambar 3.6 Sambungan Rel Kereta Api Setelah Dimodifikasi**



**Gambar 3.7 Rangkaian Sensor**

### 3.1.2 Minimum Sistem AT89S8252

Mikrokontroler yang digunakan dalam perancangan alat ini adalah IC mikrokontrol AT89S8252. Mikrokontroler dihubungkan dengan sensor, *driver* motor, *driver buzzer* dan *traffic light*.

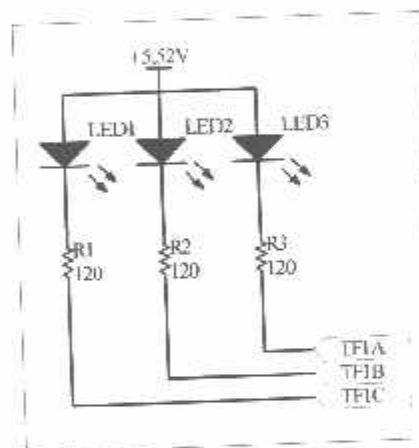
Mikrokontroller mendapat input dari sensor yang dihubungkan pada P1.0, P1.1. Kedua sensor tersebut digunakan untuk mengatur palang pintu lintasan kereta api dan *traffic light*. Dimana *driver* motor DC sebagai penggerak palang pintu lintasan kereta api yang dihubungkan pada P2.5 dan P2.6. Untuk mengatur *traffic light* digunakan P0.0 – P0.5 dan P3.0 – P3.5 karena *traffic light* menggunakan 12 lampu untuk 4 *traffic light*. Semua kontrol dan pengolahan data dilakukan oleh mikrokontroler. Gambar rangkaian secara keseluruhan dapat dilihat pada lampiran.



Adapun pengaturan nyala lampu *traffic light* dalam keadaan normal adalah sebagai berikut :

**Tabel 3.1 Pola Pengaturan Nyala Lampu Traffic Light**

<i>Traffic light</i>	1	2	3
<i>Traffic light 1</i>	Hijau	Merah	Merah
<i>Traffic light 2</i>	Merah	Hijau	Merah
<i>Traffic light 3</i>	Merah	Merah	Hijau
<i>Traffic light 4</i>	Merah	Hijau	Hijau



**Gambar 3.9 Rangkaian Traffic Light**

Analisis rangkaian *traffic light* adalah:

Diketahui tegangan sumber ( $V_{cc}$ ) adalah 5,52 V, tegangan bias led ( $V_F$ ) adalah 2,4 V dan arus ( $I_F$ ) yang mengalir pada led adalah 20 mA maka:

$$\begin{aligned}
 R_{I,ed} &= (V_{cc} - V_F) / I_F \\
 &= (5,52V - 2,4V) / 20 \times 10^{-3} \\
 &= 3,12 V / 20 \times 10^{-3} \\
 &= 150 \Omega \approx 120 \Omega
 \end{aligned}$$

### 3.1.5 Rangkaian *Driver Motor DC*

Untuk rangkaian *driver* motor DC digunakan IC ULN 2003A sebagai pengendali arah putaran mekanik motor DC. Pada IC ULN 2003A dapat dipicu dengan tegangan 12 Volt dan arus maksimum sebesar 500 mA dengan suhu kerja dari -20 C sampai 80 C. Ic ULN 2003A mampu menghidupkan dan mematikan relay yang hanya memiliki tegangan maksimal sebesar 12 Volt dengan resistansi kumparan sebesar 125,7  $\Omega$  jadi dapat diketahui arus relay sebesar :

$$\begin{aligned}
 \text{Dimana : } I_{\text{relay}} &= \frac{VCC}{R_{\text{relay}}} \\
 &= \frac{12}{125,7} = 95,4 \text{ mA}
 \end{aligned}$$

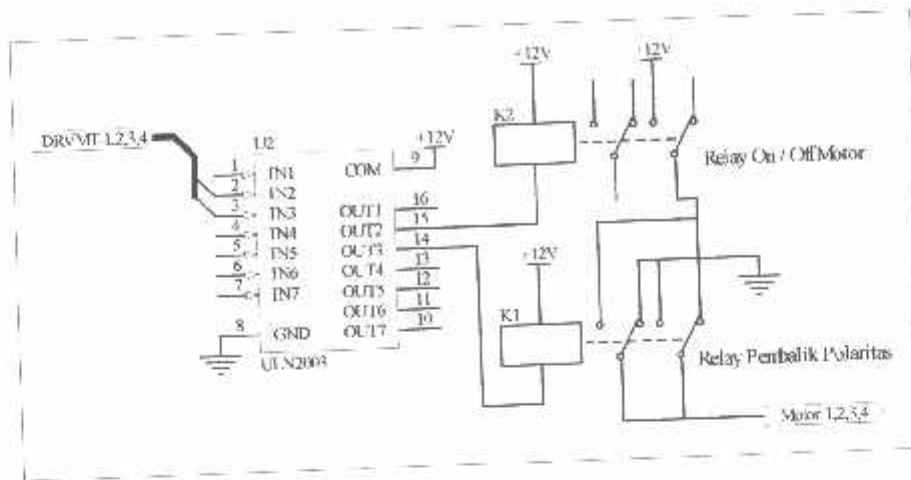
Dengan adanya arus relay sebesar 95,4 mA maka IC ULN2003A dapat menggerakkan relay tersebut karena ULN 2003A memiliki arus maksimum sebesar 500 mA sesuai dengan data sheet.

Adapun analisis motor DC adalah: Diketahui tegangan sumber ( $V_{cc}$ ) adalah 12 V, dengan resistansi 92,30  $\Omega$  jadi arus motor DC dapat diketahui sebesar:

$$\begin{aligned}
 \text{Dimana : } I_{\text{motor}} &= \frac{VCC}{R_{\text{motor}}} \\
 &= \frac{12}{92,30} = 130 \text{ mA}
 \end{aligned}$$

Di bawah ini adalah rangkaian *driver* relay motor DC :





**Gambar 3. 10 Rangkaian Driver Motor DC**

Cara kerja dari rangkaian di atas adalah sebagai berikut :

1. Apabila kaki no 2 mendapat logika *high* (1) dan kaki no 3 pada IC ULN mendapat logika *low* (0), maka pada kaki no 15 akan menjadi *low* (0) dan kaki no 14 akan menjadi *high* (1) karena terdapat gerbang not pada IC ULN 2003, Akibatnya motor akan tercatu daya sehingga motor berputar (di sini putaran motor diaplikasikan untuk menutup palang pintu lintasan kereta api).
2. Apabila kaki no 2 mendapat logika *high* (1) dan kaki no 3 pada IC ULN mendapat logika *low* (0), maka pada kaki no 14 akan menjadi *low* (0) dan kaki no 15 akan menjadi *high* (1), Akibatnya motor akan tercatu daya dengan kondisi *polaritas* berbalikan dengan kondisi pertama sehingga motor berputar dengan arah berlawanan dengan kondisi pertama (di sini putaran motor diaplikasikan untuk membuka palang pintu lintasan kereta api).

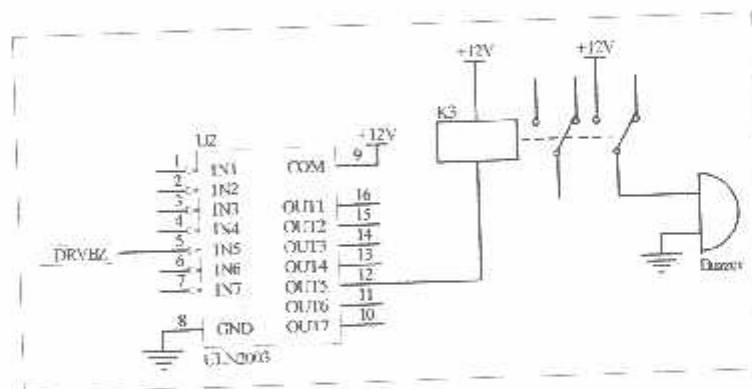
### 3.1.6 Rangkaian Driver Buzzer

Bila ada kereta api yang melintas sensor rel kereta api akan aktif, sehingga buzzer akan aktif sebagai tanda bahwa ada kereta yang melintas. Pada perencanaan alat ini IC ULN 2003A adalah IC driver yang dikemas dalam chips sebagai inputan dari buzzer.

Adapun analisis buzzer adalah: Diketahui tegangan sumber ( $V_{cc}$ ) adalah 12 V, dengan resistansi 165,7  $\Omega$  jadi arus buzzer dapat diketahui sebesar:

$$\begin{aligned}\text{Dimana : } I_{\text{buzzer}} &= \frac{V_{CC}}{R_{\text{buzzer}}} \\ &= \frac{12}{165,7} = 7,24 \text{ mA}\end{aligned}$$

Driver buzzer seperti gambar berikut :

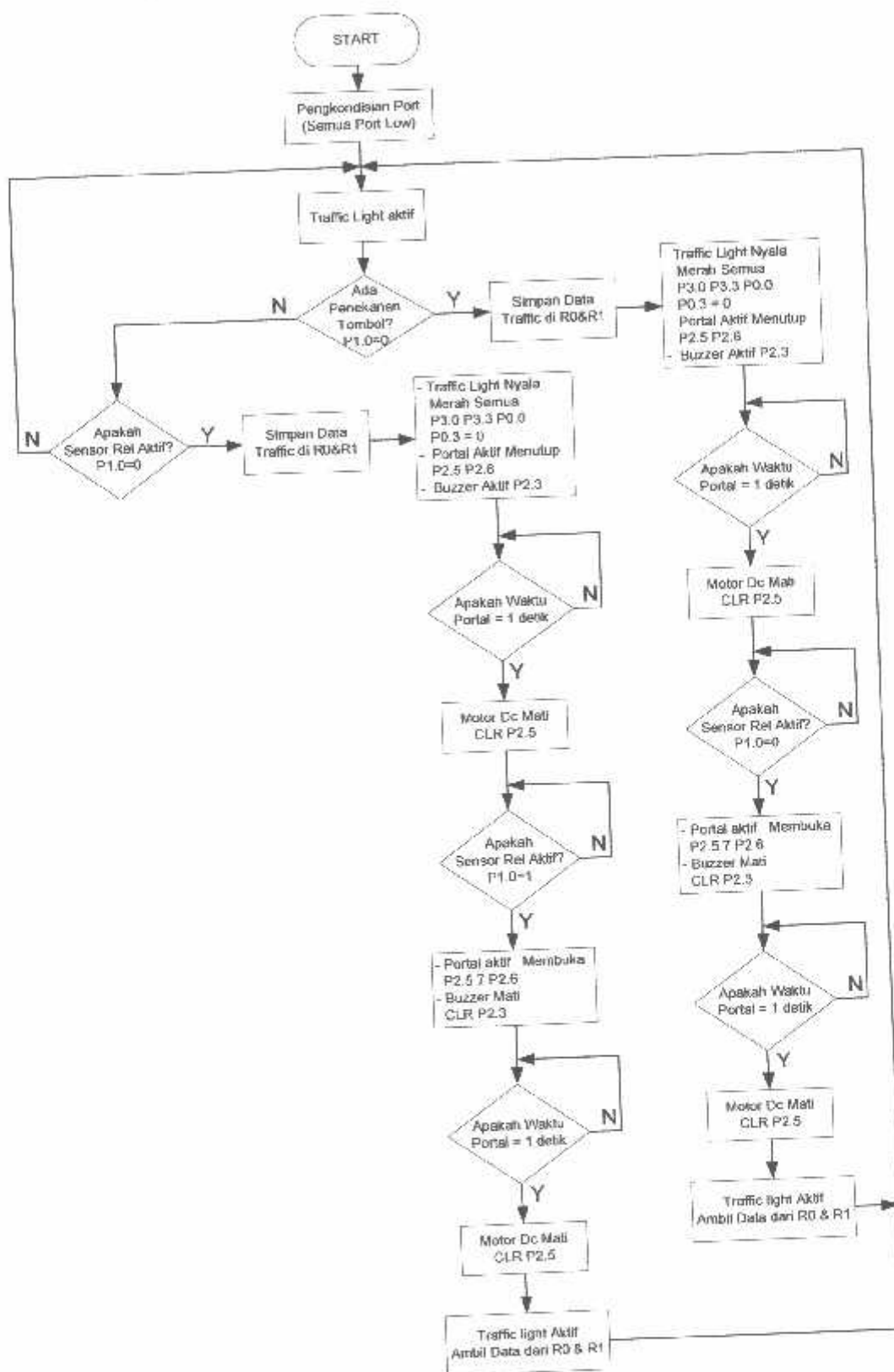


Gambar 3.11 Rangkaian Driver Buzzer

### 3.2 Perancangan Perangkat Lunak

Untuk mendukung agar perangkat keras berfungsi sesuai dengan perencanaan, maka diperlukan perangkat lunak sebagai penunjangnya. Sistem aplikasi Mikrokontroler AT89S8252 ini dapat mengatur dan mengendalikan keseluruhan sistem apabila ada urutan instruksi atau diagram alir yang mendefinisikan secara jelas urutan tugas yang harus dikerjakan.

### 3.2.1 Diagram Alir





## **BAB IV**

### **PENGUJIAN ALAT**

Pengujian alat dilakukan untuk mengetahui kinerja dari rangkaian yang sudah selesai dibuat. Pengujian alat ini menggunakan dua cara yaitu pengujian alat setiap blok dan pengujian alat secara keseluruhan dari sistem yang direncanakan. Pengujian alat setiap blok dimaksudkan untuk mempermudah dalam mencari kesalahan (*trouble shooting*) apabila rangkaian yang dibuat tidak sesuai dengan yang direncanakan. Setelah pengujian alat setiap blok selesai dan tidak ada kesalahan maka dilanjutkan dengan pengujian alat dari keseluruhan sistem yang telah direncanakan

#### **4.1 Pengujian Alat Setiap Blok**

##### **4.1.1 Pengujian Sensor Rel Kereta Api**

###### **1. Tujuan**

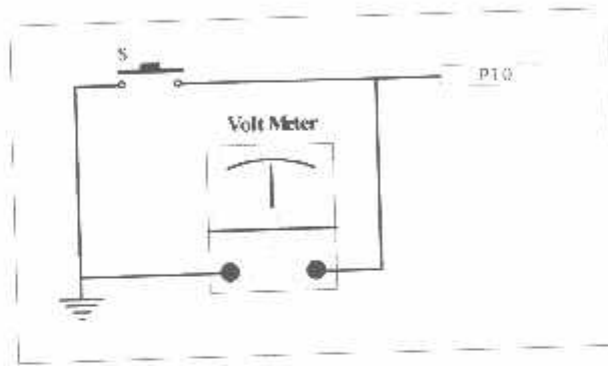
Untuk mengetahui rangkaian sensor rel kereta api apakah sudah dapat bekerja dengan baik atau belum, hal ini dapat diketahui dengan mengukur besarnya tegangan pada rangkaian tersebut pada saat sensor aktif dan tidak aktif. Prinsip kerja Sensor rel kereta api ini sama dengan switch.

###### **2. Peralatan yang digunakan**

- Multimeter digital
- Rangkaian *switch*
- Catu daya 5,52 volt

### 3. Prosedur Pengujian

- Membuat rangkaian seperti pada gambar 4.1.
- Pengukuran dilakukan dalam dua kondisi yaitu saat tidak ada penekanan dan ada penekanan.
- Memasukkan hasil pengukuran pada tabel 4.1.



**Gambar 4.1 Pengujian Sensor Rel Kereta Api**

**Tabel 4.1.**

**Hasil Pengujian Rangkaian Tombol Start**

Kondisi	Switch	
	Switch	Tegangan (Volt)
Ada Penekanan	Close	0,00 volt
Tidak Ada Penekanan	Open	5,52 volt

#### 4. Hasil pengujian :

Dari hasil pengujian rangkaian *switch* pada tabel 4.1. di atas, dapat diketahui bahwa pada saat ada penekanan ( ada kereta ) saklar akan menutup, sehingga pada P1.0 berlogika low atau "0" dan tegangannya adalah 0,00 Volt, sebaliknya pada saat kondisi tidak ada penekanan ( tidak ada kereta ) maka saklar akan terbuka sehingga P1.0 berlogika *high* atau "1" dan tegangannya adalah 5,52 Volt.

#### 4.1.2 Pengujian Driver Motor

##### 1. Tujuan:

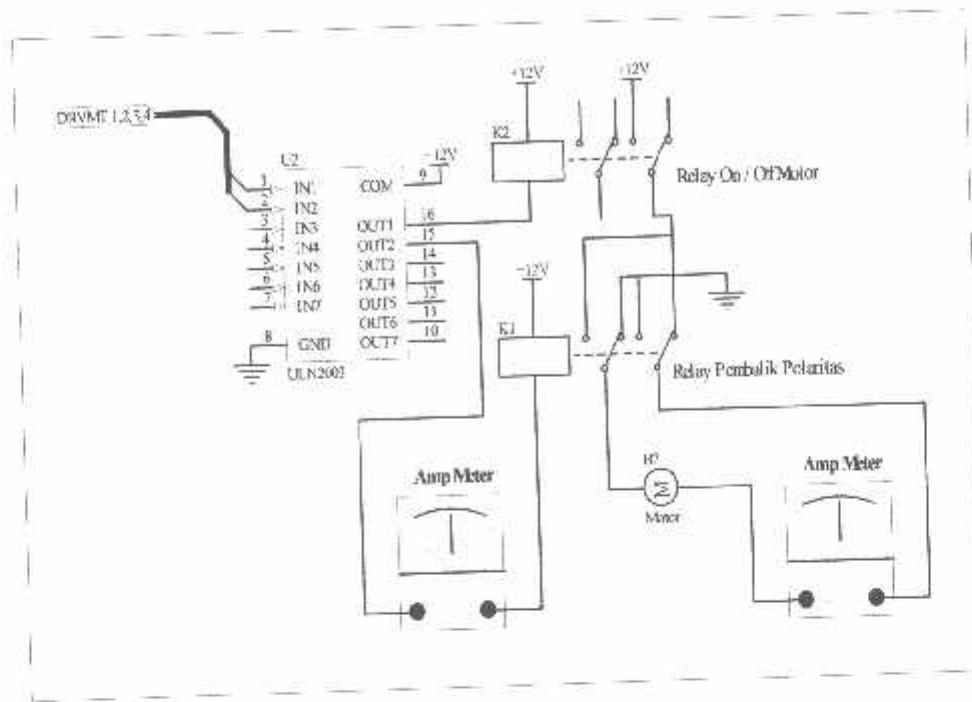
- Mengetahui besarnya arus yang mengalir pada relay dan motor.
- Mengetahui apakah driver motor sudah bekerja sesuai dengan yang direncanakan apa belum.

##### 2. Peralatan yang dibutuhkan :

- Rangkaian driver motor
- Multimeter Digital

##### 3. Prosedur pengujian :

- Buat rangkaian seperti pada gambar 4.2
- Rangkaian diberi catu daya 12 volt
- Ukur besar arus yang mengalir pada relay dan motor
- Mencatat hasil pengukuran.



Gambar 4.2 Pengujian Rangkaian *Driver Relay*

#### 4. Hasil Pengujian

Tabel 4.2 Hasil Pengukuran Arus Pada Motor

No	P2.5	P2.6	Kondisi Palang Pintu	Arus Motor		$\Delta I$ (mA)	Error (%)
				Perhitungan	Pengukuran		
1	0	0	Diam	0 mA	0 mA	0	0
2	1	0	Sedang Menutup	130 mA	133 mA	3	2,3
3	0	1	Sedang Membuka	130 mA	133 mA	3	2,3



**Tabel 4.3 Hasil Pengukuran Arus Pada Relay**

No	P2.5	P2.6	Kondisi Palang Pintu	Arus Relay		AI (mA)	Error (%)
				Perhitungan	Pengukuran		
1	0	0	Diam	0 mA	0 mA	0	0
2	1	0	Sedang Menutup	95,4 mA	93,3 mA	2,1	2,2
3	0	1	Sedang Membuka	95,4 mA	93,3 mA	2,1	2,2

Dari hasil pengujian rangkaian driver motor, besarnya arus yang mengalir pada saat kondisi relay aktif adalah 93,3 mA, sedangkan arus yang mengalir pada saat motor aktif adalah 133 mA. Apabila P2.5 diberi logika "1" dan P2.6 diberi logika "0" maka palang pintu kereta api dalam keadaan menutup sedangkan apabila P2.5 diberi logika "0" dan P2.6 diberi logika "1" maka palang pintu kereta api dalam keadaan membuka. Sehingga dari hasil pengujian maka rangkaian driver motor tersebut dapat bekerja sesuai dengan perencanaan.

#### 4.1.3 Pengujian Rangkaian Buzzer

##### 1. Tujuan:

- Mengetahui rangkaian buzzer sudah dapat berfungsi dengan baik atau belum dan mengukur besarnya arus yang mengalir pada buzzer.

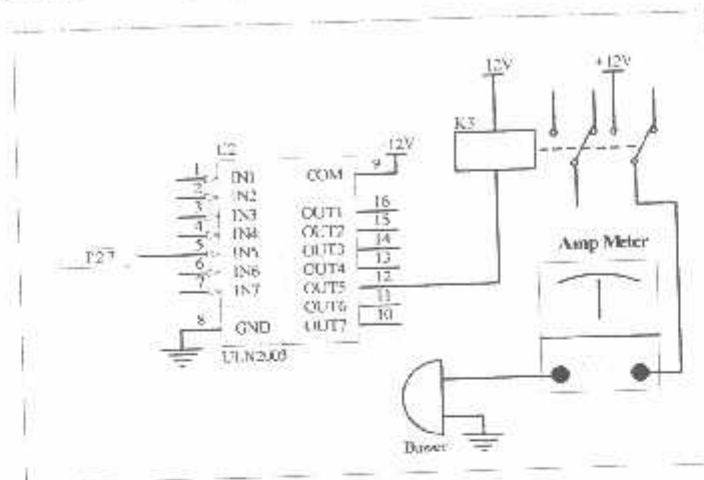
##### 2. Peralatan yang dibutuhkan :

- Rangkaian buzzer
- Avo meter digital

##### 3. Prosedur pengujian :

- Buat rangkaian seperti pada gambar 4.3

- Rangkaian diberi catu daya 12 volt
- Mencatat hasil pengukuran.



**Gambar 4.3 Pengujian Rangkaian Buzzer**

#### 4. Hasil Pengujian

**Tabel 4.4 Hasil Pengukuran Arus Pada Rangkaian Buzzer**

No	P2.5	Kondisi Buzzer	Arus Buzzer		$\Delta I$ (mA)	Error (%)
			Perhitungan	Pengukuran		
1	0	Berbunyi	0 mA	0 mA	0	0
2	1	Tidak Berbunyi	7,24 mA	7,08 mA	0,16	2,2

Dari hasil pengujian rangkaian buzzer, besarnya arus yang mengalir pada saat kondisi buzzer berbunyi aktif adalah 7,08 mA, sedangkan arus yang mengalir pada saat buzzer tidak berbunyi aktif adalah 0,00 mA. Apabila P2.3 diberi logika "0" maka buzzer dalam keadaan Berbunyi sedangkan apabila P2.3 diberi logika "1" maka buzzer dalam keadaan

tidak berbunyi. Sehingga dari hasil pengujian maka rangkaian buzzer tersebut dapat berfungsi sesuai dengan perencanaan.

#### 4.1.4 Pengujian Traffic Light

1. Tujuan:

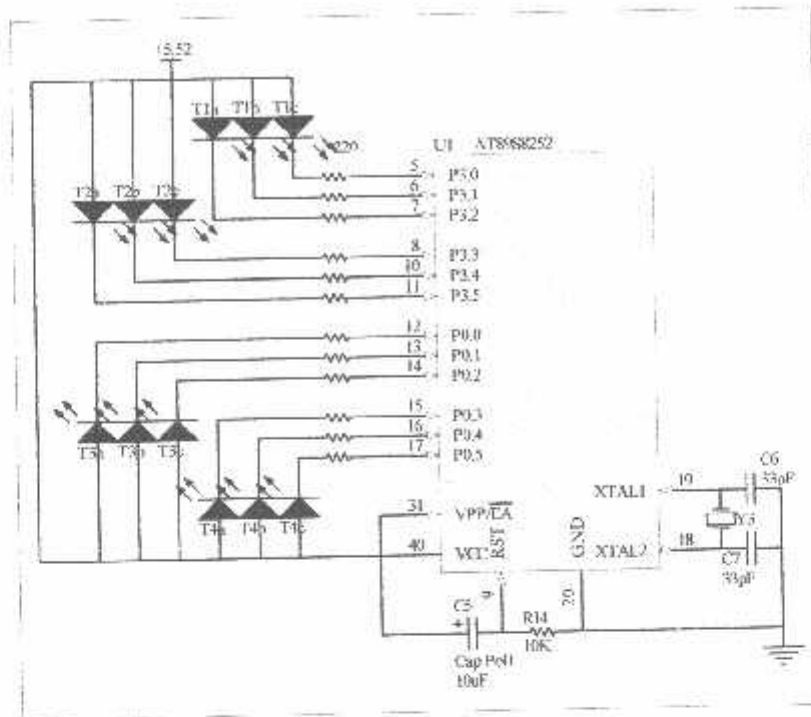
- Mengetahui selisih waktu delay *traffic light* antara pengukuran dengan program.

2. Peralatan yang dibutuhkan :

- Rangkaian traffic light
- Stopwatch

3. Prosedur pengujian :

- Buat rangkaian seperti pada gambar 4.4
- Rangkaian diberi catu daya 5,52 volt
- Buat program untuk menjalankan traffic light
- Kemudian jalankan program
- Mencatat hasil pengukuran.



**Gambar 4.4 Pengujian Traffic Light**

#### 4. Hasil Pengujian

Nyala lampu traffic light adalah sebagai berikut:

- Hijau ke Kuning
- Kuning ke Merah
- Merah ke Hijau

Sehingga dari lampu hijau menyala sampai kembali ke hijau lagi dalam program ditentukan selama 72 detik dengan kondisi traffic light menyala sesuai dengan pola yang ditentukan.

PUTAR:

SETB	P3.0	
SETB	P3.1	
CLR	P3.2	;T1 HIDUP HIJAU
CLR	P3.3	;T2 HIDUP MERAH
SETB	P3.4	
SETB	P3.5	

CLR	P0.6	;T3 HIDUP MERAH
SETB	P0.1	
SETB	P0.2	

CLR	P0.3	;T4 HIDUP MERAH
SETB	P0.4	
SETB	P0.5	

CALL	TUNDA_3_DT
CALL	TUNDA_3_DT
CALL	TUNDA_3_DT
CALL	TUNDA_3_DT
CALL	TUNDA_3_DT
CALL	TUNDA_3_DT

SETB	P3.0
CLR	P3.1
SETB	P3.2

SETB	P3.3
SETB	P3.4
CLR	P3.5

CLR	P0.6
SETB	P0.1
SETB	P0.2

SETB	P0.3
SETB	P0.4
CLR	P0.5

CALL	TUNDA_3_DT
CALL	TUNDA_3_DT
SETB	P3.1
CLR	P3.0
CALL	TUNDA_3_DT
CALL	TUNDA_3_DT
CALL	TUNDA_3_DT
CALL	TUNDA_3_DT
CALL	TUNDA_3_DT

CLR	P3.0
SETB	P3.1
SETB	P3.2

SETB	P3.3
------	------

```

CLR      P3.4
SETB     P3.5

SETB     P0.6
SETB     P0.1
CLR      P0.2

SETB     P0.3
SETB     P0.4
CLR      P0.5

CALL     TUNDA_3_DT
CALL     TUNDA_3_DT
CLR      P3.3
SETB     P3.4
CALL     TUNDA_3_DT
CALL     TUNDA_3_DT
CALL     TUNDA_3_DT
CALL     TUNDA_3_DT
CALL     TUNDA_3_DT
CALL     TUNDA_3_DT
CALL     TUNDA_3_DT

SETB     P0.2
SETB     P0.5
CLR      P0.1
CLR      P0.4
CALL     TUNDA_3_DT
CALL     TUNDA_3_DT
SETB     P0.1
SETB     P0.4
AJMP     PUTAR

```

**Tabel 4.5 Pola Pengaturan Nyala Lampu Traffic Light**

<i>Traffic light</i>	1	2	3
<i>Traffic light 1</i>	Hijau	Merah	Merah
<i>Traffic light 2</i>	Merah	Hijau	Merah
<i>Traffic light 3</i>	Merah	Merah	Hijau
<i>Traffic light 4</i>	Merah	Hijau	Hijau

Perhitungan waktu delay pada traffic light:

- Waktu untuk 1 kali instruksi atau 1 siklus mesin adalah:

$$\begin{aligned}
 1 \text{ Siklus mesin} &= \frac{1}{f_{Osc}/12} \\
 &= \frac{1}{\frac{12MHz}{12}} \\
 &= \frac{12}{12} \\
 &= 1.10^{-6} = 1\mu S
 \end{aligned}$$

- Program delay *traffic light*:

	MOV	R7,#100	;Isi Register R7 dengan 100
LOOP2:	MOV	R6,#100	;Isi Register R6 dengan 100
LOOP1:	MOV	R5,#100	;Isi Register R5 dengan 100
	DJNZ	R5,\$	;Kurangi Register R5 sampai = 0
	JB	P1.0,TOMBOL1a	;Periksa P1.0 Bila = 1 maka lompat keTOMBOL1a
	CALL	ADA_KERETA	;Panggil Subrutin ADA_KERETA
TOMBOL1a:	DJNZ	R6,LOOP1	;Kurangi Register R6 jika Belum = 0 Maka lompat ke LOOP1
	JB	P1.0,TOMBOL1b	;Periksa P1.0 Bila = 1 maka lompat keTOMBOL1b
	CALL	ADA_KERETA	;Panggil Subrutin ADA_KERETA
TOMBOL1b:	DJNZ	R7,LOOP2	;Kurangi Register R7 jika Belum = 0 Maka lompat ke LOOP2
	JB	P1.0,TOMBOL1	;Periksa P1.0 Bila = 1 maka lompat ke TOMBOL1
	CALL	ADA_KERETA	;Panggil Subrutin ADA_KERETA
TOMBOL1:	JB	P1.1,LANGSUNG1	;Periksa P1.0 Bila = 1 maka lompat ke TOMBOL1
	CALL	KERETA_DARI_STASIUN	

Dalam program R5,R6,R7 masing-masing diisi 100, sehingga setiap pengurangan Register 5,6,7 sampai = 0 dibutuhkan waktu:  $(100 \times 100 \times 100) \times 1\mu S = 1 \text{ s}$

Dan untuk Tunda\_3\_detik =  $1 \text{ s} \times 3 = 3 \text{ s}$

Jadi waktu delay yang dibutuhkan *traffic light* dari lampu hijau menyala sampai kembali ke hijau lagi adalah: Tunda\_3\_Detik  $\times 24 = 72 \text{ detik}$

Kemudian dilakukan pengukuran sebanyak 5 kali, waktu perpindahan nyala lampu traffic light dari lampu hijau menyala sampai kembali ke hijau lagi dengan stopwatch didapatkan data sebagai berikut:

**Tabel 4.6 Hasil Pengujian Traffic Light**

No	Hasil Perhitungan	Hasil Pengukuran
1	72 detik	70,63 detik
2	72 detik	70,98 detik
3	72 detik	71,08 detik
4	72 detik	71,21 detik
5	72 detik	71,38 detik
Jumlah	360 detik	355,23 detik



Dari hasil pengujian tersebut diperoleh rata-rata error sebesar:

$$\begin{aligned}\text{Kesalahan (error)} &= \frac{\text{Perhitungan} - \text{Pengukuran}}{\text{Perhitungan}} \times 100\% \\ &= \frac{360 \text{ detik} - 355,23 \text{ detik}}{360 \text{ detik}} \times 100\% \\ &= 1,32 \%\end{aligned}$$

#### 4.2 Pengujian Sistem Secara Keseluruhan

Pengujian seluruh sistem dilakukan dengan membandingkan hasil rancangan dengan standar perancangan awal.

##### 1. Tujuan :

Untuk mengetahui apakah sistem yang telah dibuat dapat bekerja sesuai dengan perencanaan. Pengujian dilakukan dengan menjalankan kereta api pada jalur simulasi yang sudah dibuat.

##### 2. Peralatan yang digunakan:

- Jalur yang telah dibuat seperti pada gambar di bawah
- Simulasi kereta api
- Rangkaian alat secara keseluruhan.

##### 3. Proses Pengujian:

- Menjalankan kereta api pada jalur simulasi yang telah dibuat.
- Kemudian mengamati sistem kerja dari seluruh rangkaian.

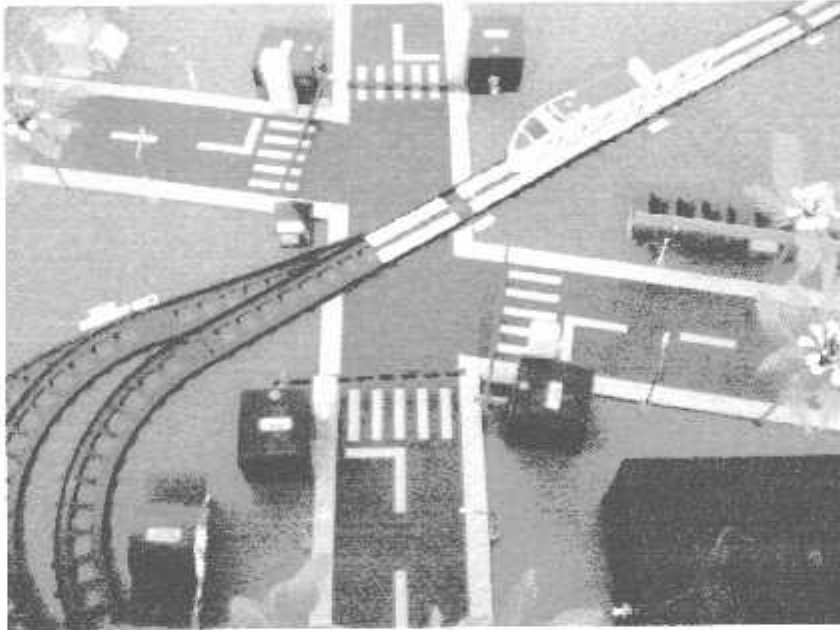
#### 4. Hasil pengujian :

Berdasarkan pengujian yang telah dilakukan terlihat bahwa pada saat kereta api melintas dan menyentuh detektor maka *buzzer* berbunyi, *traffic light* menyala merah semua dan palang pintu lintasan kereta juga menutup semua. Kemudian setelah kereta api melewati detektor maka *buzzer* tidak aktif, *traffic light* kembali menyala normal dengan mengambil data terakhir pada saat sebelum *traffic light* menyala merah semua dan palang pintu lintasan kereta api juga membuka kembali.

#### 4.3 Spesifikasi alat:

1. Nama alat : simulasi palang pintu lintasan kereta api dan *traffic light* secara otomatis.
2. Fungsi Alat : untuk mempermudah mengetahui kedatangan kereta api dan menggantikan sistem kerja alat yang manual menjadi otomatis.
3. Keistimewaan atau kemampuan tambahan alat adalah penutupan palang pintu lintasan kereta api yang disesuaikan juga dengan nyala lampu *traffic light*.
4. Kinerja Alat:
  - Sistem penggerak palang pintu lintasan kereta api menggunakan motor DC
  - *Traffic light* menggunakan lampu led dengan warna: hijau, kuning, merah.
  - Sumber daya menggunakan tegangan 12 Volt dan 5,5 Volt.

5. Dimensi jalur sebagai simulasi adalah
  - Jalur terbuat dari bahan plastik yang dilapisi dengan plat besi.
6. Simulasi alat yang telah dibuat dapat dilihat pada gambar berikut:



**Gambar 4.5 Simulasi Alat**



## BAB V

### PENUTUP

Pada bab ini penulis akan menyampaikan kesimpulan dan saran setelah melakukan pengujian pada alat simulasi palang pintu perlintasan kereta api dan traffic light secara otomatis.

#### 5.1 Kesimpulan

Setelah melakukan perencanaan, pembuatan, dan pengujian pada alat simulasi palang pintu perlintasan kereta api dan traffic light secara otomatis, dapat diambil beberapa kesimpulan sebagai berikut :

1. Detektor rel kereta api pada alat ini bekerja sesuai dengan sistem kerja pada switch, karena itu dapat diketahui bahwa pada saat ada *penekanan* (kereta lewat ) saklar akan menutup, sehingga pada P1.0 terjadi logika low atau "0" dan tegangannya adalah 0,00 Volt, sebaliknya pada saat kondisi *tidak ada penekanan* ( tidak ada kereta ) maka saklar akan terbuka sehingga pada P1.0 logika *high* atau "1" dan tegangannya adalah 5,52 Volt.
2. Pada rangkaian driver motor, besarnya arus yang mengalir pada saat kondisi relay aktif adalah 93,3 mA, sedangkan arus yang mengalir pada saat motor aktif adalah 133 mA. Karena itu apabila P2.5 diberi logika "1" dan P2.6 diberi logika "0" maka palang pintu kereta api dalam keadaan menutup sedangkan apabila P2.5 diberi logika "1" dan P2.6 diberi logika

“1” maka palang pintu kereta api dalam keadaan membuka. Sehingga didapatkan error motor sebesar 2,3 % dan error relay sebesar 2,2 %

3. Pada rangkaian buzzer, besarnya arus yang mengalir pada saat kondisi buzzer berbunyi adalah 7,08 mA, dan arus yang mengalir pada saat buzzer tidak berbunyi adalah 0,00 mA. Apabila P2.3 diberi logika “0” maka buzzer berbunyi sedangkan apabila P2.3 diberi logika “1” maka buzzer tidak berbunyi. Sehingga didapatkan error buzzer sebesar 2,2 %
4. Waktu perpindahan nyala lampu traffic light dari lampu hijau menyala sampai kembali ke hijau lagi dalam program ditentukan selama 72 detik dengan kondisi traffic light menyala sesuai dengan pola yang ditentukan. Kemudian dilakukan pengukuran sebanyak 5 kali, waktu perpindahan nyala lampu traffic light dari lampu hijau menyala sampai kembali ke hijau lagi dengan stopwatch didapatkan error sebesar 1,32 %

## 5.2 Saran

Dalam skripsi ini masih terdapat beberapa kekurangan. Untuk mencapai hasil yang lebih sempurna dapat diberikan saran-saran sebagai berikut :

1. Agar sistem ini dapat bekerja dengan baik harus dilakukan pengecekan berkala pada jalur rel kereta api, karena jalur rel kereta api memiliki peranan sangat penting dalam kinerja alat ini, yaitu sebagai pendeteksi apabila ada kereta api yang akan melintas.
2. Kendala utama system ini adalah pada saat musim hujan, karena curah hujan yang tinggi dapat mengganggu kinerja system ini terutama pada detector rel

kereta api. Untuk mengatasi kendala tersebut kita harus dapat mengatur peletakan rel kereta api dengan baik agar tidak terendam oleh air.

3. Sistem akan bekerja dengan baik (stabil) jika digunakan komponen-komponen yang berkualitas tinggi.
4. Untuk mengatasi apabila listrik PLN mati perlu digunakan catu daya lain sebagai *back-up power supply*.





## Daftar Pustaka

AT89C8252. American: Atmel Corporation. 2000

Budioko, Totok, *Belajar Dengan Mudah Dan Cepat Pemrograman Bahasa C Dengan SDCC Pada Mikrokontroler AT89X051/AT89C51/52*, Gava Media: Yogyakarta, 2005.

Malik, Ibnu, *Belajar Mikrokontroller ATMEL*. Gava Media: Yogyakarta, 2003.

Paul Malvino, Albert; *Prinsip-Prinsip Elektronika*, Salemba Teknika: Jakarta, 2003.

Robert F Coughlin, Fedrerick F Driscoll, *Penguat Operasional Dan Rangkaian Terpadu*, Erlangga: Jakarta, 1994.

Website ATMEL Corp. (<http://www.atmel.com>)

---





**INSTITUT TEKNOLOGI NASIONAL MALANG**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**JURUSAN TEKNIK ELEKTRO S-1**  
**KONSENTRASI T.ELEKTRONIKA**

**FORMULIR PERBAIKAN SKRIPSI**

Nama : Juli Surya Purwanto  
Nim : 02.17.017  
Masa Bimbingan : 10 Oktober 2006 s/d 10 April 2007  
Judul : Perancangan dan Pembuatan Simulasi Palang Pintu Kereta  
Api dan "Traffic Light" di Kota Lama Malang Secara  
Otomatis Berbasis Mikrokontroler AT89S8252

No	Tanggal	Meteri Perbaikan	Paraf Penguji I
1.	22-03-2007	<ul style="list-style-type: none"><li>- Batasan Masalah</li><li>- Pengujian Sensor Rel Kereta Api</li><li>- Pengujian-pengujian yang Lain</li><li>- Hasil Perhitungan Tabel 4.6</li></ul>	

Mengetahui

Dosen Pembimbing I

Dosen Pembimbing II

( Ir. Widodo Pudji Mulyanto, MT. )

( I Komang Somawirata, ST. )

NIP.Y 1028700171

NIP.1030100361

Disetujui

Dosen Penguji I

( Joseph Dedy Irawan, ST., MT. )  
NIP.132315178



**INSTITUT TEKNOLOGI NASIONAL MALANG**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**JURUSAN TEKNIK ELEKTRO S-1**  
**KONSENTRASI T.ELEKTRONIKA**

**FORMULIR PERBAIKAN SKRIPSI**

Nama : Juli Surya Purwanto  
Nim : 02.17.017  
Masa Bimbingan : 10 Oktober 2006 s/d 10 April 2007  
Judul : Perancangan dan Pembuatan Simulasi Palang Pintu Kereta  
Api dan "Traffic Light" di Kota Lama Malang Secara  
Otomatis Berbasis Mikrokontroller AT89S8252

No	Tanggal	Meteri Perbaikan	Paraf Penguji I
1.	21-03-2007	<ul style="list-style-type: none"><li>- Kesimpulan</li><li>- Kondisi Pengujian Cuaca Hujan</li><li>- Daftar Pustaka</li></ul>	

Mengetahui

Dosen Pembimbing I

( Ir. Widodo Pudji Mulyanto, MT. )

NIP.Y 1028700171

Dosen Pembimbing II

( I Komang Somawirata, ST. )

NIP.1030100361

Disetujui

Dosen Penguji I

(M. Ashar,ST.,MT)



INSTITUT TEKNOLOGI NASIONAL  
JL. BENDUNGAN SIGURA-GURA 2  
MALANG

---

## LEMBAR BIMBINGAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI


- |                                  |  |
|----------------------------------|--|
| 1. Nama Mahasiswa                | : Juli Surya Purwanto  |
| 2. NIM                           | : 02.17.017  |
| 3. Jurusan                       | : Teknik Elektronika S-1   |
| 4. Program Studi                 | : Teknik Elektro   |
| 5. Judul Skripsi                 | : Perancangan dan Pembuatan<br>Simulasi Palang Pintu Kereta<br>Api dan "Traffic Light" di Kota<br>Lama Malang Secara Otomatis<br>Berdasarkan Mikrokontroler<br>AT89S8252 |
| 6. Tanggal Mengajukan Skripsi    | : 10 Oktober 2006  |
| 7. Selesai Menulis Skripsi       | : 10 April 2007  |
| 8. Dosen Pembimbing I            | : Ir. Widodo Pudji Mulyanto, MT  |
| 9. Telah Dievaluasi Dengan Nilai | : 80 (Delapan Puluh)   |

**Mengetahui**  
**Ketua Jurusan Teknik Elektro**



**(Ir. F. Yudi limpraptono, MT)**  
**NIP.Y 1039500274**

**Malang, Maret 2007**  
**Dosen Pembimbing I**



**(Ir. Widodo Pudji Mulyanto, MT)**  
**NIP.Y 1028700171**

---



INSTITUT TEKNOLOGI NASIONAL  
JL. BENDUNGAN SIGURA-GURA 2  
MALANG

---

## LEMBAR BIMBINGAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

- |                                  |  |
|----------------------------------|--|
| 1. Nama Mahasiswa                | : Juli Surya Purwanto  |
| 2. NIM                           | : 02.17.017  |
| 3. Jurusan                       | : Teknik Elektronika S-1   |
| 4. Program Studi                 | : Teknik Elektro   |
| 5. Judul Skripsi                 | : Perancangan dan Pembuatan<br>Simulasi Palang Pintu Kereta<br>Api dan "Traffic Light" di Kota<br>Lama Malang Secara Otomatis<br>Berdasarkan Mikrokontroler<br>AT89S8252 |
| 6. Tanggal Mengajukan Skripsi    | : 10 Oktober 2006  |
| 7. Selesai Menulis Skripsi       | : 10 April 2007  |
| 8. Dosen Pembimbing II           | : I Komang Somawirata, ST.MT.  |
| 9. Telah Dievaluasi Dengan Nilai | : 90 (Sembilan Puluh)  |

**Mengetahui**  
**Ketua Jurusan Teknik Elektro**

  
**(Ir.F. Yudi limpraptono, MT)**  
**NIP.Y 1039500274**

**Malang, Maret 2007**  
**Dosen Pembimbing I**

  
**( I Komang Somawirata, ST.MT. )**  
**NIP.1030100361**

---



### FORMULIR BIMBINGAN SKRIPSI

Nama : Juli Surya Purwanto  
NIM : 02.17.017  
Masa Bimbingan : 10 Oktober 2006 s/d 10 April 2007  
Judul Skripsi : Perancangan Dan Pembuatan Simulasi Palang Pintu Kereta Api dan "TRAFFIC LIGHT" Di Kota Lama Malang Secara Otomatis Berbasis Mikrokontroller AT89S8252

No	Tanggal	Uraian	Paraf Pembimbing
1	16/07/06	Konsultasi Bab I, II, III	
2	24/07/06	Acce Bab I, II, III	
3	31/07/06	Revisi Bab IV (Gambar)	
4	12/07/02	Makalah Seminar Hasil	
5	28/07/02	konsultasi Laporan	
6	7/10/07	Acc Nya	
7			
8			
9			
10			

Malang,

Dosen Pembimbing I

Ir. Widodo Pudi Mulyanto.MT.  
NIP.Y.1028700171



### FORMULIR BIMBINGAN SKRIPSI

Nama : Juli Surya Purwanto  
NIM : 02.17.017  
Masa Bimbingan : 10 Oktober 2006 s/d 10 April 2007  
Judul Skripsi : Perancangan Dan Pembuatan Simulasi Palang Pintu Kereta Api dan "TRAFFIC LIGHT" Di Kota Lama Malang Secara Otomatis Berbasis Mikrokontroler AT89S8252

No	Tanggal	Uraian	Paraf Pembimbing
1	16/06/06	Konsultasi awal proposal.	
2	16/07/06	Konsultasi BAB I, BAB II, BAB III	
3	17/07/06	Revisi BAB II	
4	21/07/06	Revisi Bab IV & Gambar akhir.	
5	28/07/06	Revisi Bab IV & Bab V	
6	11/08/06	Mahasiswa Seminar hasil	
7	28/08/06	Konsultasi skripsi	
8	29/08/06	SKRIPSI	
9			
10			

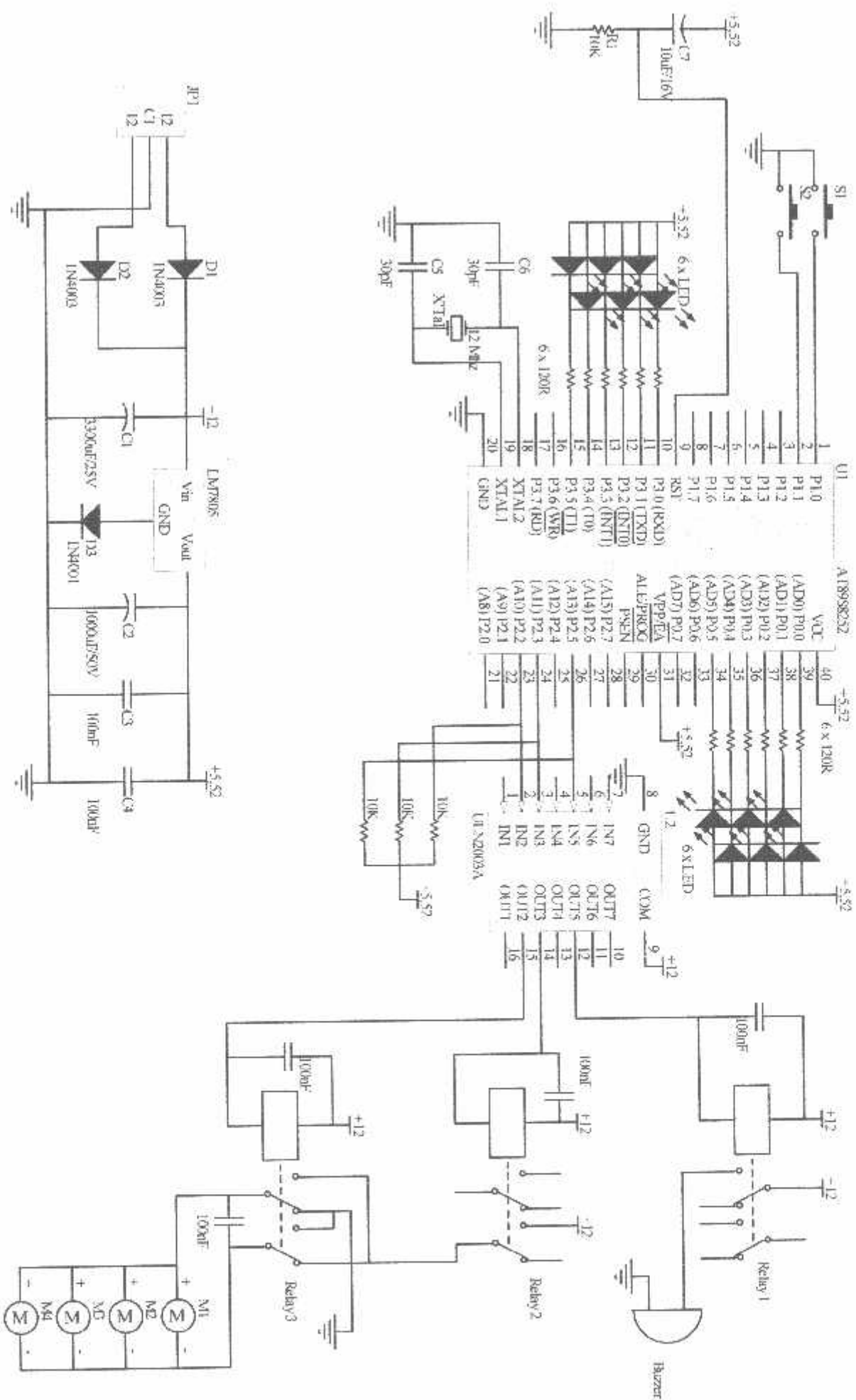
Malang, 8-03-2007

Dosen Pembimbing II

I Komang Somawirata, ST.  
NIP.1030100361

Form. S - 4b





```
org 0000H
JMP START
```

RT:

```
CLR P2.7
CLR P2.6
CLR P2.5
CLR P2.4
CLR P2.3
CLR P2.2
CLR P2.1
CLR P2.0
```

\*\*\*\*\*

RT:

```
SETB P3.0      ;P3.0 diberi logika 1 (T1 Lampu Merah Tidak Menyala)
SETB P3.1      ;P3.1 diberi logika 1 (T1 Lampu kuning Tidak Menyala)
CLR P3.2       ;P3.2 diberi logika 0 (T1 Lampu Hijau Menyala)

CLR P3.3       ;P3.3 diberi logika 0 (T2 Lampu Merah Menyala)
SETB P3.4      ;P3.4 diberi logika 1 (T2 Lampu kuning Tidak Menyala)
SETB P3.5      ;P3.5 diberi logika 1 (T2 Lampu Hijau Tidak Menyala)

CLR P0.6       ;P0.6 diberi logika 0 (T3 Lampu Merah Menyala)
SETB P0.1      ;P0.1 diberi logika 1 (T3 Lampu kuning Tidak Menyala)
SETB P0.2      ;P0.2 diberi logika 1 (T3 Lampu Hijau Tidak Menyala)

CLR P0.3       ;P0.3 diberi logika 0 (T4 Lampu Merah Menyala)
SETB P0.4      ;P0.4 diberi logika 1 (T4 Lampu kuning Tidak Menyala)
SETB P0.5      ;P0.5 diberi logika 1 (T4 Lampu Hijau Tidak Menyala)

CALL TUNDA_3_DT ;Panggil Subrutin Tunda_3_DT
CALL TUNDA_3_DT ;Panggil Subrutin Tunda_3_DT
CALL TUNDA_3_DT ;Panggil Subrutin Tunda_3_DT
CALL TUNDA_3_DT ;Panggil Subrutin Tunda_3_DT
CALL TUNDA_3_DT ;Panggil Subrutin Tunda_3_DT
CALL TUNDA_3_DT ;Panggil Subrutin Tunda_3_DT

SETB P3.0      ;P3.0 diberi logika 1 (T1 Lampu Merah Tidak Menyala)
CLR P3.1       ;P3.1 diberi logika 0 (T1 Lampu kuning Menyala)
SETB P3.2      ;P3.2 diberi logika 1 (T1 Lampu Hijau Tidak Menyala)

SETB P3.3      ;P3.3 diberi logika 1 (T2 Lampu Merah Tidak Menyala)
SETB P3.4      ;P3.4 diberi logika 1 (T2 Lampu kuning Tidak Menyala)
CLR P3.5       ;P3.5 diberi logika 0 (T2 Lampu Hijau Menyala)

CLR P0.6       ;P0.6 diberi logika 0 (T3 Lampu Merah Menyala)
SETB P0.1      ;P0.1 diberi logika 1 (T3 Lampu kuning Tidak Menyala)
SETB P0.2      ;P0.2 diberi logika 1 (T3 Lampu Hijau Tidak Menyala)
```

---

SETB	P0.3	;P0.3 diberi logika 1 (T4 Lampu Merah Tidak Menyala)
SETB	P0.4	;P0.4 diberi logika 1 (T4 Lampu Kuning Tidak Menyala)
CLR	P0.5	;P0.5 diberi logika 0 (T4 Lampu Hijau Menyala)
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
SETB	P3.1	;P3.1 diberi logika 1 (T1 Lampu Kuning Tidak Menyala)
CLR	P3.0	;P3.0 diberi logika 0 (T1 Lampu Merah Menyala)
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CLR	P3.0	;P3.0 diberi logika 0 (T1 Lampu Merah Menyala)
SETB	P3.1	;P3.1 diberi logika 1 (T1 Lampu Kuning Tidak Menyala)
SETB	P3.2	;P3.2 diberi logika 1 (T1 Lampu Hijau Tidak Menyala)
SETB	P3.3	;P3.3 diberi logika 1 (T2 Lampu Merah Tidak Menyala)
CLR	P3.4	;P3.4 diberi logika 0 (T2 Lampu Kuning Menyala)
SETB	P3.5	;P3.5 diberi logika 1 (T2 Lampu Hijau Tidak Menyala)
SETB	P0.6	;P0.6 diberi logika 1 (T3 Lampu Merah Tidak Menyala)
SETB	P0.1	;P0.1 diberi logika 1 (T3 Lampu Kuning Tidak Menyala)
CLR	P0.2	;P0.2 diberi logika 0 (T3 Lampu Hijau Menyala)
SETB	P0.3	;P0.3 diberi logika 1 (T4 Lampu Merah Tidak Menyala)
SETB	P0.4	;P0.4 diberi logika 1 (T4 Lampu Kuning Tidak Menyala)
CLR	P0.5	;P0.5 diberi logika 0 (T4 Lampu Hijau Menyala)
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CLR	P3.3	;P3.3 diberi logika 0 (T2 Lampu Merah Menyala)
SETB	P3.4	;P3.4 diberi logika 1 (T2 Lampu Kuning Tidak Menyala)
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
SETB	P0.2	;P0.2 diberi logika 1 (T3 Lampu Hijau Tidak Menyala)
SETB	P0.5	;P0.5 diberi logika 1 (T4 Lampu Hijau Tidak Menyala)
CLR	P0.1	;P0.1 diberi logika 0 (T3 Lampu Kuning Menyala)
CLR	P0.4	;P0.4 diberi logika 0 (T4 Lampu Kuning Menyala)
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
CALL	TUNDA_3_DT	;Panggil Subrutin Tunda_3_DT
SETB	P0.1	;P0.1 diberi logika 1 (T3 Lampu Kuning Tidak Menyala)

---

```

SETB    P0.4      ;P0.4 diberi logika 1 (T4 Lampu kuning Tidak Menyala)
AJMP     PUTAR     ;Lompat ke Putar

```

#### ETA\_DART\_STASIUN:

```

MOV      R2,P3      ;Pindahkan isi P3 ke Register R2
MOV      R1,P0      ;Pindahkan isi P0 ke Register R1
CLR      P3.0       ;P3.0 diberi logika 0 (T1 Lampu Merah Menyala)
SETB     P3.1       ;P3.1 diberi logika 1 (T1 Lampu kuning Tidak Menyala)
SETB     P3.2       ;P3.2 diberi logika 1 (T1 Lampu Hijau Tidak Menyala)

CLR      P3.3       ;P3.3 diberi logika 0 (T2 Lampu Merah Menyala)
SETB     P3.4       ;P3.4 diberi logika 1 (T2 Lampu kuning Tidak Menyala)
SETB     P3.5       ;P3.5 diberi logika 1 (T2 Lampu Hijau Tidak Menyala)

CLR      P0.6       ;P0.6 diberi logika 0 (T3 Lampu Merah Menyala)
SETB     P0.1       ;P0.1 diberi logika 1 (T3 Lampu kuning Tidak Menyala)
SETB     P0.2       ;P0.2 diberi logika 1 (T3 Lampu Hijau Tidak Menyala)

CLR      P0.3       ;P0.3 diberi logika 0 (T4 Lampu Merah Menyala)
SETB     P0.4       ;P0.4 diberi logika 1 (T4 Lampu kuning Tidak Menyala)
SETB     P0.5       ;P0.5 diberi logika 1 (T4 Lampu Hijau Tidak Menyala)

SETB     P2.3       ;P2.3 diberi logika 1 (Duzzet Aktif)
CALL     ROLL       ;Panggil Subrutin ROLL
SETB     P2.6       ;P2.6 diberi logika 1 (Palang Pintu Lintasan Kereta Api

```

Stop)

```

ACALL    ROLL       ;Panggil Subrutin ROLL
SETB     P2.5       ;P2.5 diberi logika 1 (Motor Aktif)
CALL     TUNDA_POR   ;Panggil Subrutin TUNDA_POR
CLR      P2.5       ;P2.5 diberi logika 0 (Motor Tidak Aktif)
SETB     P2.3       ;P2.3 diberi logika 1 (Buzzer Aktif)

SETB     P2.3       ;P2.3 diberi logika 1 (Buzzer Aktif)
JB       P1.0,PUT     ;Periksa P1.0 Bila = 1 maka lompat ke PUT
CLR      P2.3       ;P2.3 diberi logika 0 (Buzzer Tidak Aktif)
CALL     ROLL       ;Panggil Subrutin ROLL
CLR      P2.6       ;P2.6 diberi logika 0 (Palang Pintu Lintasan Kereta Api

```

uka)

```

ACALL    ROLL       ;Panggil Subrutin ROLL
SETB     P2.5       ;P2.5 diberi logika 1 (Motor Aktif)
CALL     TUNDA_POR   ;Panggil Subrutin TUNDA_POR
CLR      P2.5       ;P2.5 diberi logika 0 (Motor Tidak Aktif)
MOV      P3,R2      ;Pindahkan isi Register R2 ke P3
MOV      P0,R1      ;Pindahkan isi Register R1 ke P0

```

:

```

JNB      P1.0,PUT4   ;Periksa P1.0 Bila = 0 maka lompat ke PUT4
CALL     ROLL       ;Panggil Subrutin ROLL
JNB      P1.0,PUT4   ;Periksa P1.0 Bila = 0 maka lompat ke PUT4
RET                          ;Kembali ke program Utama

```

## KERETA:

```

MOV    R2,P3      ;Pindahkan isi P3 ke Register R2
MOV    R1,P0      ;Pindahkan isi P0 ke Register R1

CLR    P3.0       ;P3.0 diberi logika 0 (T1 Lampu Merah Menyala)
SETB   P3.1       ;P3.1 diberi logika 1 (T1 Lampu kuning Tidak Menyala)
SETB   P3.2       ;P3.2 diberi logika 1 (T1 Lampu Hijau Tidak Menyala)

CLR    P3.3       ;P3.3 diberi logika 0 (T2 Lampu Merah Menyala)
SETB   P3.4       ;P3.4 diberi logika 1 (T2 Lampu kuning Tidak Menyala)
SETB   P3.5       ;P3.5 diberi logika 1 (T2 Lampu Hijau Tidak Menyala)

CLR    P0.6       ;P0.6 diberi logika 0 (T3 Lampu Merah Menyala)
SETB   P0.1       ;P0.1 diberi logika 1 (T3 Lampu kuning Tidak Menyala)
SETB   P0.2       ;P0.2 diberi logika 1 (T3 Lampu Hijau Tidak Menyala)

CLR    P0.3       ;P0.3 diberi logika 0 (T4 Lampu Merah Menyala)
SETB   P0.4       ;P0.4 diberi logika 1 (T4 Lampu kuning Tidak Menyala)
SETB   P0.5       ;P0.5 diberi logika 1 (T4 Lampu Hijau Tidak Menyala)

SETB   P2.3       ;P2.3 diberi logika 1 (Buzzer Aktif)
CALL   ROLL       ;Panggil Subrutin ROLL
SETB   P2.6       ;P2.6 diberi logika 1 (Palang Pintu Lintasan Kereta Api)

```

Lap):

```

CALL   ROLL       ;Panggil Subrutin ROLL
SETB   P2.5       ;P2.5 diberi logika 1 (Motor Aktif)
CALL   TUNDA_POR  ;Panggil Subrutin TUNDA_POR
CLR    P2.5       ;P2.5 diberi logika 0 (Motor Tidak Aktif)

```

:

```

SETB   P2.3       ;P2.3 diberi logika 1 (Buzzer Aktif)
JNB    P1.0,P0T1  ;Periksa P1.0 Bila = 0 maka lompat ke P0T1
CALL   ROLL       ;Panggil Subrutin ROLL
JNB    P1.0,P0T1  ;Periksa P1.0 Bila = 0 maka lompat ke P0T1
CALL   ROLL       ;Panggil Subrutin ROLL
JNB    P1.0,P0T1  ;Periksa P1.0 Bila = 0 maka lompat ke P0T1
CALL   ROLL       ;Panggil Subrutin ROLL
CLR    P2.3       ;P2.3 diberi logika 0 (Buzzer Tidak Aktif)
CALL   ROLL       ;Panggil Subrutin ROLL
CLR    P2.6       ;P2.6 diberi logika 0 (Palang Pintu Lintasan Kereta Api)

```

uka):

```

CALL   ROLL       ;Panggil Subrutin ROLL
SETB   P2.5       ;P2.5 diberi logika 1 (Motor Aktif)
CALL   TUNDA_POR  ;Panggil Subrutin TUNDA_POR
CLR    P2.5       ;P2.5 diberi logika 0 (Motor Tidak Aktif)
MOV    P3,R2      ;Pindahkan isi Register R2 ke P3
MOV    P0,R1      ;Pindahkan isi Register R1 ke P0
RET                     ;Kembali ke program Utama

```

\*\*\*\*\*

A\_3 DT:

```

MOV    R7,#100      ;Isi Register R7 dengan 100
P2: MOV    R6,#100      ;Isi Register R6 dengan 100
P1: MOV    R5,#100      ;Isi Register R5 dengan 100
    DJNZ   R5,S        ;Kurangi Register R5 sampai = 0
    JB     P1.0,TOMBOLA ;Periksa P1.0 Bila = 1 maka lompat ke TOMBOLA
    CALL   ADA_KERETA   ;Panggil Subrutin ADA_KERETA
BOLA:
    DJNZ   R6,LOOP1     ;Kurangi Register R6 jika Belum = 0 Maka lompat ke LOOP1
    JB     P1.0,TOMBOLA ;Periksa P1.0 Bila = 1 maka lompat ke TOMBOLA
    CALL   ADA_KERETA   ;Panggil Subrutin ADA_KERETA
BOLB:
    DJNZ   R7,LOOP2     ;Kurangi Register R7 jika Belum = 0 Maka lompat ke LOOP2

    JB     P1.0,TOMBOLA ;Periksa P1.0 Bila = 1 maka lompat ke TOMBOLA
    CALL   ADA_KERETA   ;Panggil Subrutin ADA_KERETA
OL1:
    JB     P1.1,LANGSUNG1 ;Periksa P1.0 Bila = 1 maka lompat ke TOMBOLA
    CALL   KERETA_DARI_STASIUN ;Panggil Subrutin KERETA_DARI_STASIUN
;SUNG1:

MOV    R7,#100      ;Isi Register R7 dengan 100
21: MOV    R6,#100      ;Isi Register R6 dengan 100
11: MOV    R5,#100      ;Isi Register R5 dengan 100
    DJNZ   R5,S        ;Kurangi Register R5 sampai = 0
    JB     P1.0,TOMBOLA ;Periksa P1.0 Bila = 1 maka lompat ke TOMBOLA
    CALL   ADA_KERETA   ;Panggil Subrutin ADA_KERETA
OL1C:
    DJNZ   R6,LOOP11    ;Kurangi Register R6 jika Belum = 0 Maka lompat ke LOOP11
    JB     P1.0,TOMBOLA ;Periksa P1.0 Bila = 1 maka lompat ke TOMBOLA
    CALL   ADA_KERETA   ;Panggil Subrutin ADA_KERETA
OL1D:
    DJNZ   R7,LOOP21    ;Kurangi Register R7 jika Belum = 0 Maka lompat ke LOOP21
*****
    JB     P1.0,TOMBOLA2 ;Periksa P1.0 Bila = 1 maka lompat ke TOMBOLA2
    CALL   ADA_KERETA   ;Panggil Subrutin ADA_KERETA
OL2:
    JB     P1.1,LANGSUNG2 ;Periksa P1.1 Bila = 1 maka lompat ke LANGSUNG2
    CALL   KERETA_DARI_STASIUN ;Panggil KERETA_DARI_STASIUN
SUNG2:

MOV    R7,#100      ;Isi Register R7 dengan 100
22: MOV    R6,#100      ;Isi Register R6 dengan 100
12: MOV    R5,#100      ;Isi Register R5 dengan 100
    DJNZ   R5,S        ;Kurangi Register R5 sampai = 0
    JB     P1.0,TOMBOLA ;Periksa P1.0 Bila = 1 maka lompat ke TOMBOLA
    CALL   ADA_KERETA   ;Panggil Subrutin ADA_KERETA
OL2A:
    DJNZ   R6,LOOP12    ;Kurangi Register R6 jika Belum = 0 Maka lompat ke LOOP12
    JB     P1.0,TOMBOLA ;Periksa P1.0 Bila = 1 maka lompat ke TOMBOLA
    CALL   ADA_KERETA   ;Panggil Subrutin ADA_KERETA

```

---

```

BOV1f:
    DJNZ    R7,LOOP22      ;Kurangi Register R7 jika Belum = 0 Maka lompat ke LOOP22

    JB      P1.0,TOMBOL3   ;Periksa P1.0 Bila = 1 maka lompat ke TOMBOL3
    CALL    ADA_KERETA     ;Panggil Subrutin ADA_KERETA

BOV3:
    JB      P1.1,LANGSUNG3 ;Periksa P1.1 Bila = 1 maka lompat ke LANGSUNG3
    CALL    KERETA_DARI_STASIUN ;Panggil KERETA DARI STASIUN

LANGSUNG3:
    RET                      ;Kembali ke program Utama
*****

DA POR:
    MOV     R7,#100        ;Isi Register R7 dengan 100
10: MOV     R6,#100        ;isi Register R6 dengan 100
9:  MOV     R5,#100        ;Isi Register R5 dengan 100
    DJNZ    R5,$           ;Kurangi Register R5 sampai = 0
    DJNZ    R6,LOOP9       ;Kurangi Register R6 jika Belum = 0 Maka lompat ke LOOP9
    DJNZ    R7,LOOP10      ;Kurangi Register R7 jika Belum = 0 Maka lompat ke LOOP10
    RET

11:
    MOV     R7,#50         ;Isi Register R7 dengan 50
AC: MOV     R6,#50         ;Isi Register R6 dengan 50
AD: MOV     R5,#100        ;Isi Register R5 dengan 100
    DJNZ    R5,$           ;Kurangi Register R5 sampai = 0
    DJNZ    R6,LOOPAD      ;Kurangi Register R6 jika Belum = 0 Maka lompat ke LOOPAD
    DJNZ    R7,LOOPAC      ;Kurangi Register R7 jika Belum = 0 Maka lompat ke LOOPAC
    RET                  ;Kembali ke program Utama

END

```

## Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
- SPI Serial Interface for Program Downloading
- Endurance: 1,000 Write/Erase Cycles
- 1K Bytes EEPROM
- Endurance: 100,000 Write/Erase Cycles
- 5V Operating Range
- Static Operation: 0 Hz to 24 MHz
- Two-Level Program Memory Lock
- 8-bit Internal RAM
- 8 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Multiple Interrupt Sources
- Programmable UART Serial Channel
- Serial Interface
- Power Idle and Power-down Modes
- SPI Recovery From Power-down
- Programmable Watchdog Timer
- Data Pointer
- Power-off Flag

## Description

AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of downloadable Flash programmable and erasable read only memory and 1K bytes of EEPROM. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 pinout. The on-chip downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a traditional nonvolatile memory programmer. By combining a versatile 8-bit CPU with 8K bytes of downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

AT89S8252 provides the following standard features: 8K bytes of downloadable Flash, 1K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode allows the CPU to be shut down while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but disables the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from. Once lock Bit 2 has been activated.



## 8-bit Microcontroller with 8K Bytes Flash

**AT89S8252**

Rev. 0401E-02/00



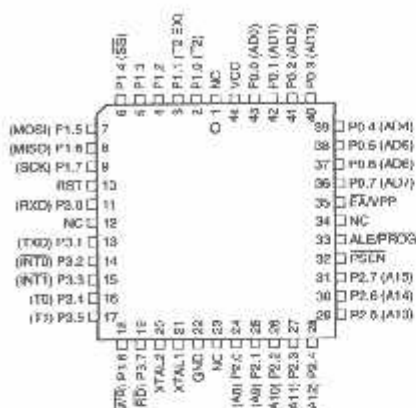


## Configurations

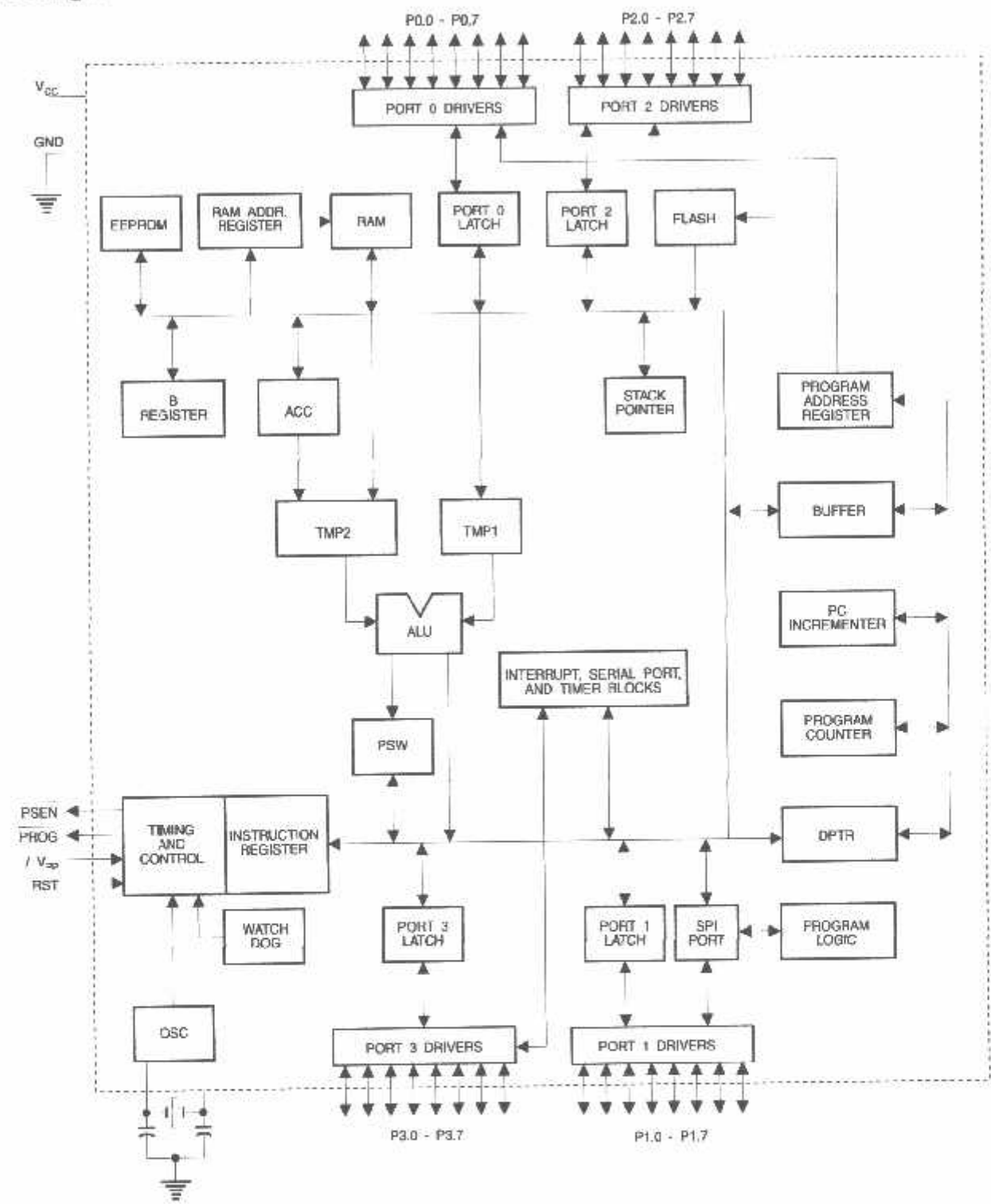
PDIP



PLCC



k Diagram



Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

## Description

more, P1.4, P1.5, P1.6, and P1.7 can be configured SPI slave port select, data input/output and shift put/output pins as shown in the following table.

n	Alternate Functions
	T2 (external count input to Timer/Counter 2), clock-out
	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
	SS (Slave port select input)
	MOSI (Master data output, slave data input pin for SPI channel)
	MISO (Master data input, slave data output pin for SPI channel)
	SCK (Master clock output, slave clock input pin for SPI channel)

also receives the low-order address bytes during programming and verification.

is an 8-bit bi-directional I/O port with internal pullups. When 2 output buffers can sink/source four TTL inputs. When data is written to Port 2 pins, they are pulled high by internal pullups and can be used as inputs. As inputs, pins that are externally being pulled low will source  $I_{IL}$  because of the internal pullups.

emits the high-order address byte during fetches external program memory and during accesses to data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-up emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ R1), Port 2 outputs the contents of the P2 Special Function Register.

Also receives the high-order address bits and some signals during Flash programming and verification.

is an 8 bit bi-directional I/O port with internal pullups. When 3 output buffers can sink/source four TTL inputs. When data is written to Port 3 pins, they are pulled high by internal pullups and can be used as inputs. As inputs,

Port 3 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

## RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

## ALE/PROG

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location BEH. With the bit set, ALE is active only during a MOVX or MOVX instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

## PSEN

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

## EA/VPP

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external pro-

memory locations starting at 0000H up to FFFFH. However, that if lock bit 1 is programmed, EA will be latched on reset.

It should be strapped to  $V_{CC}$  for internal program execution. This pin also receives the 12-volt programming voltage ( $V_{PP}$ ) during Flash programming when 12-volt programming is selected.

#### XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

#### XTAL2

Output from the inverting oscillator amplifier.

AT89S8252 SFR Map and Reset Values

								0FFH
B 00000000								0F7H
								0EFH
ACC 00000000								0E7H
								0DFH
PSW 00000000					SPCR 000001XX			0D7H
T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
								0C7H
IP XX000000								0BFH
P3 11111111								0B7H
IE 0X000000		SPSR 00XXXXXX						0AFH
P2 11111111								0A7H
SCON 00000000	SBUF XXXXXXXX							9FH
P1 11111111						WMCON 00000010		97H
TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H



## Special Function Registers

of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Accesses to these addresses will in general return data, and write accesses will have an indeterminate

software should not write 1s to these unlisted

locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

**Timer 2 Registers** Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

### T2CON—Timer/Counter 2 Control Register

Address = 0C8H

Reset Value = 0000 0000B

Writable

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
7	6	5	4	3	2	1	0

Bit	Function
7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
5	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
4	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
3	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
1	Timer or counter select for Timer 2. C/T2 = 0 for timer function. C/T2 = 1 for external event counter (falling edge triggered).
0	Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

**dog and Memory Control Register** The WMCON register contains control bits for the Watchdog Timer (shown in Table 3). The EEMEN and EEMWE bits are used

to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

### 1. WMCON—Watchdog and Memory Control Register

ON Address = 96H

Reset Value = 0000 0010B

PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
7	6	5	4	3	2	1	0

Bit	Function
7:5	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.
4	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.
3	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.
2	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1.
1:0	Watchdog Timer Reset and EEPROM Ready/Busy Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/BSY flag in a Read-Only mode during EEPROM write. RDY/BSY = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/BSY bit equals "0" and is automatically reset to "1" when programming is completed.
0	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.

**Serial Peripheral Interface Registers** Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data during serial data transfer sets the Write Collision Flag, WCOL, in the SPSR register. The SPDR is double buffered and the values in SPDR are not changed by

**Interrupt Registers** The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In the IP register, the individual interrupt enable bit for the SPI is in the IP register. Two priorities can be set for each of the interrupt sources in the IP register.

**Dual Data Pointer Registers** To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR WMCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

**Power Off Flag** The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.





## I. SPCR—SPI Control Register

Address = D5H

Reset Value = 0000 01XXB

SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
7	6	5	4	3	2	1	0

Bit	Function
7	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.
6	SPI Enable. SPI = 1 enables the SPI channel and connects $\overline{SS}$ , MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.
5	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.
4	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.
3	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.
2	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.
1, 0	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, $F_{osc}$ , is as follows: SPR1SPR0 SCK = $F_{osc}$ divided by 0 0 4 0 1 16 1 0 64 1 1 128

## SPSR – SPI Status Register

Address = AAH

Reset Value = 00XX XXXXB

SPIF	WCOL	–	–	–	–	–	–
7	6	5	4	3	2	1	0

Bit	Function
7	SPI Interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.
6	Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.

## SPDR – SPI Data Register

Address = 86H

Reset Value = unchanged

SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
7	6	5	4	3	2	1	0

AT89S8252

## Memory – EEPROM and RAM

89S8252 implements 2K bytes of on-chip EEPROM for storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the addresses as the SFR space but are physically separate SFR space.

In instruction accesses an internal location above 0x7F, the address mode used in the instruction is whether the CPU accesses the upper 128 bytes or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses SFR at location 0A0H (which is P2).

```
MOV P2, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses a byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

On-chip EEPROM data memory is selected by setting the  $\overline{AEN}$  bit in the WMCON register at SFR address 96H. The EEPROM address range is from 000H to 0FFH. The MOVX instructions are used to access the M. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EEMWE bit in the WMCON register needs to be set to "1" to enable any byte location in the EEPROM can be written. Software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the programming mode are self-timed and typically take 4 to 8 ms. The progress of EEPROM write can be monitored using the RDY/BSY bit (read-only) in SFR WMCON.  $\overline{RDY} = 0$  means programming is still in progress and  $\overline{RDY} = 1$  means EEPROM write cycle is completed and the write cycle can be initiated.

When, during EEPROM programming, an attempted write to the EEPROM will fetch the byte being written to the MSB complemented. Once the write cycle is complete, the data are valid at all bit locations.

## Programmable Watchdog Timer

Programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1, and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the

actual timer periods (at  $V_{CC} = 5V$ ) are within  $\pm 30\%$  of the nominal.

The WDT is disabled by Power-on Reset and during Power-down. It is enabled by setting the WDTE bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			Period (nominal)
PS2	PS1	PS0	
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

## Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

## Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which

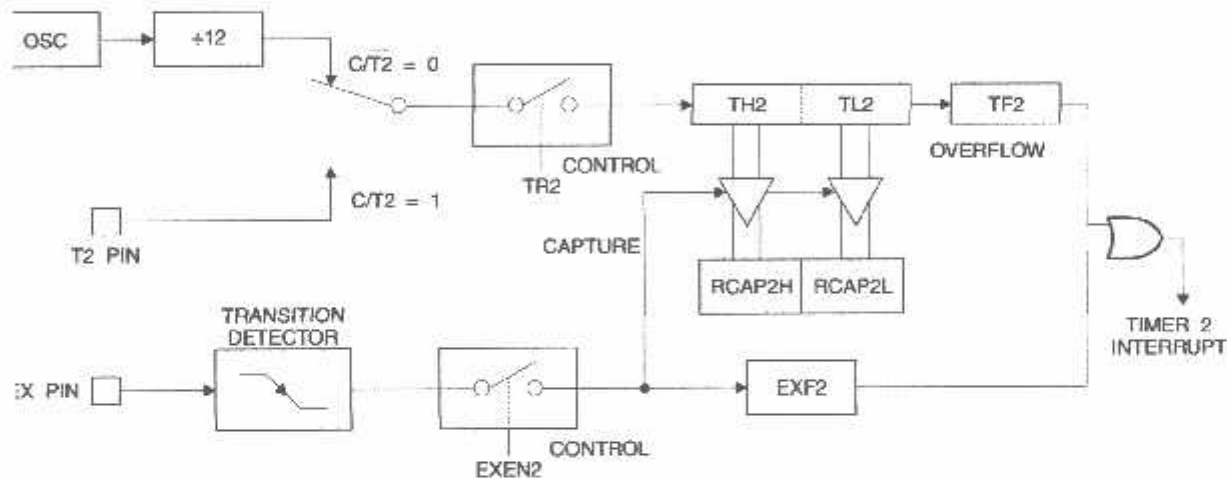


transition was detected. Since two machine cycles (24 or periods) are required to recognize a 1-to-0 transition the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least before it changes, the level should be held for at least one machine cycle.

Timer 2 Operating Modes

+ TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
1	X	0	(Off)

Timer 2 in Capture Mode



### Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an Interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

### reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when enabled in its 16 bit auto-reload mode. This feature is controlled by the DCEN (Down Counter Enable) bit located in the T2MOD register (see Table 9). Upon reset, the DCEN bit is 0 so that timer 2 will default to count up. When the DCEN bit is set, Timer 2 can count up or down, depending on the logic level of the T2EX pin.

Figure 3 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit 1 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0xFFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a bit reload can be triggered either by an overflow or

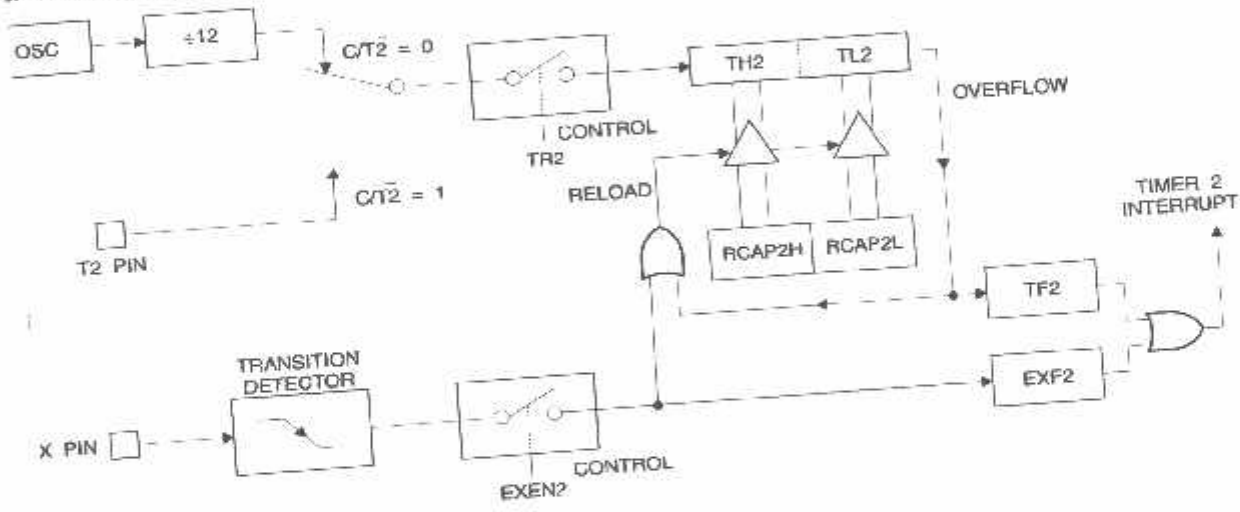
by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0xFFFFH and set the TF2 bit. This overflow also causes the 16 bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0xFFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

#### 2. Timer 2 in Auto Reload Mode (DCEN = 0)



T2MOD – Timer 2 Mode Control Register

Reset Value = XXXX XX00B

Address = 0C9H

Addressable

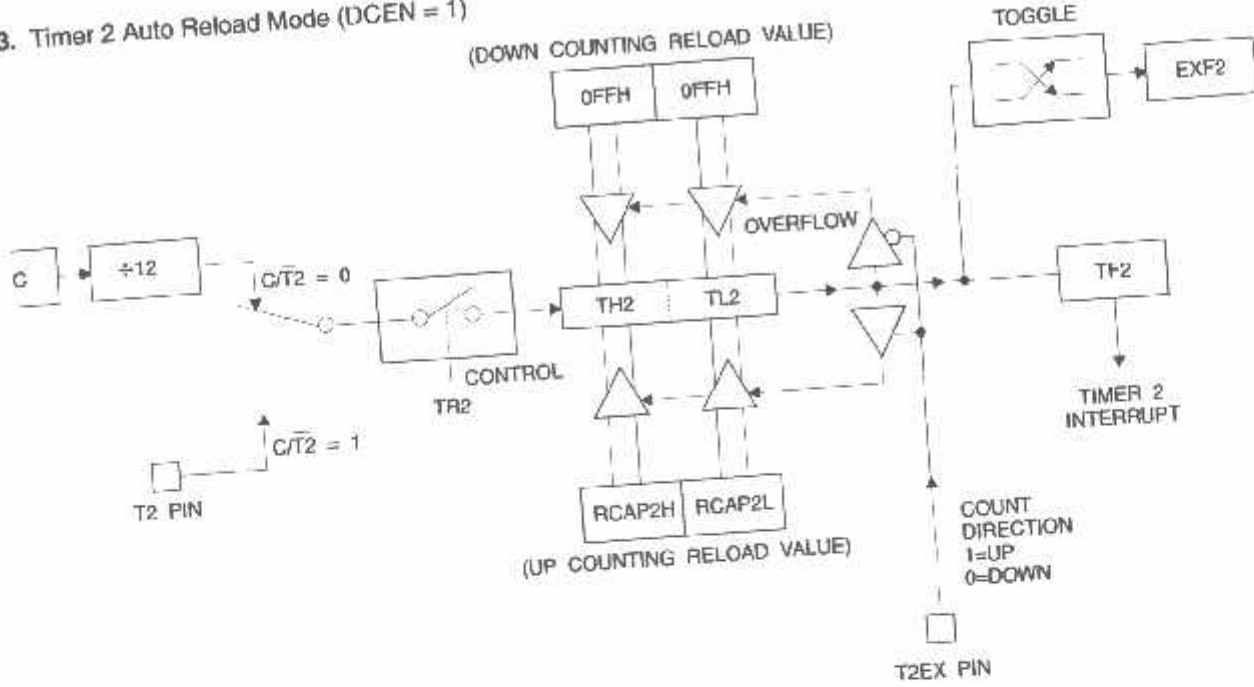
7	6	5	4	3	2	1	0
						T2OE	DCEN

#### Function

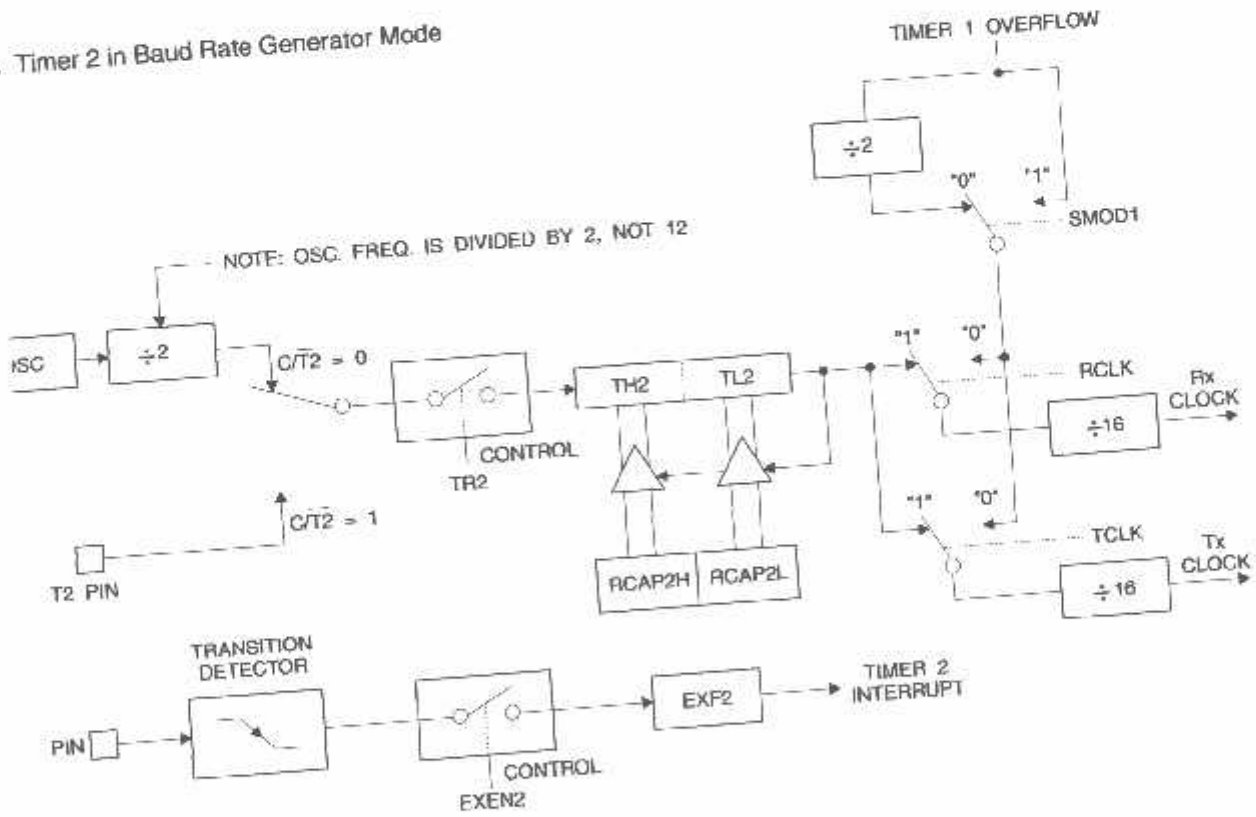
- Not implemented, reserved for future use.
- Timer 2 Output Enable bit.
- When set, this bit allows Timer 2 to be configured as an up/down counter.



### 3. Timer 2 Auto Reload Mode (DCEN = 1)



### i. Timer 2 in Baud Rate Generator Mode



## Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting RCLK and/or TCLK in T2CON (Table 2). Note that the rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

Baud rate generator mode is similar to the auto-reload mode. A rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software.

Baud rates in Modes 1 and 3 are determined by Timer 2 overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

Timer 2 can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/12 = 0). The timer operation is different from when it is used as a baud rate generator. Normally, Timer 2 increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

$$\text{Modes 1 and 3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (RCAP2H, RCAP2L)]}$$

(RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L, taken as a 16 bit unsigned integer.

When used as a baud rate generator is shown in Figure 4. This mode is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload of RCAP2H, RCAP2L to (TH2, TL2). Thus when Timer

2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer 2 is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

## Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

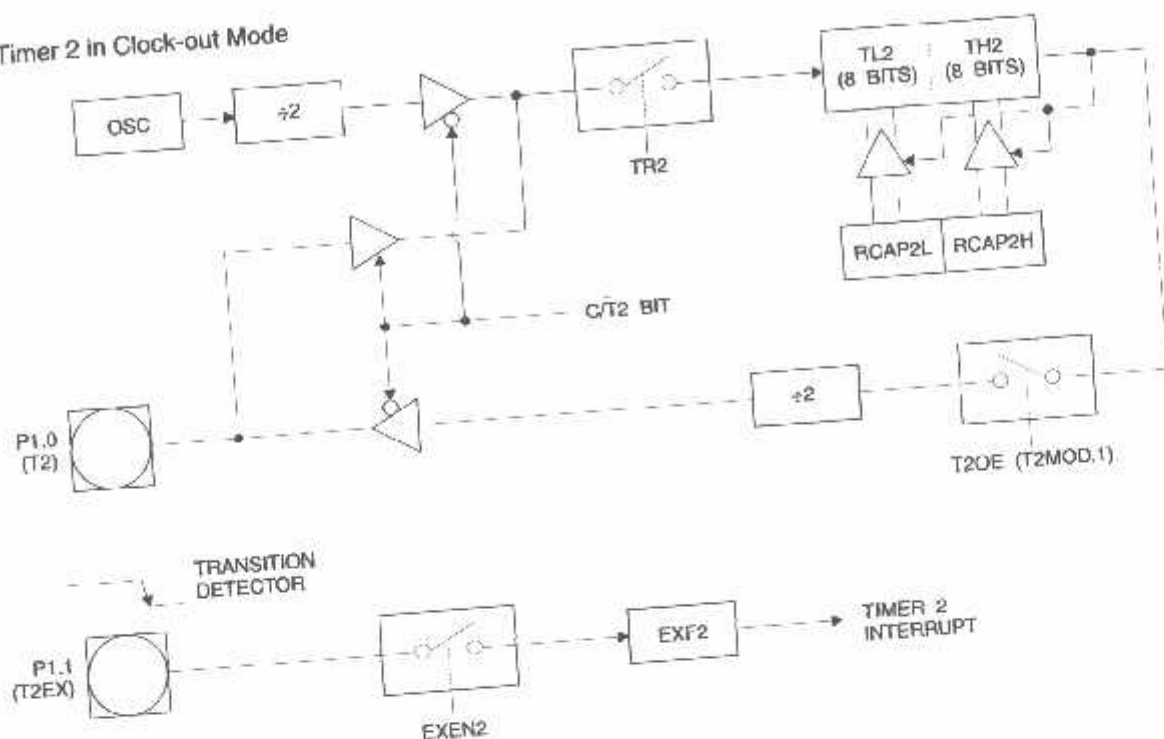
To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

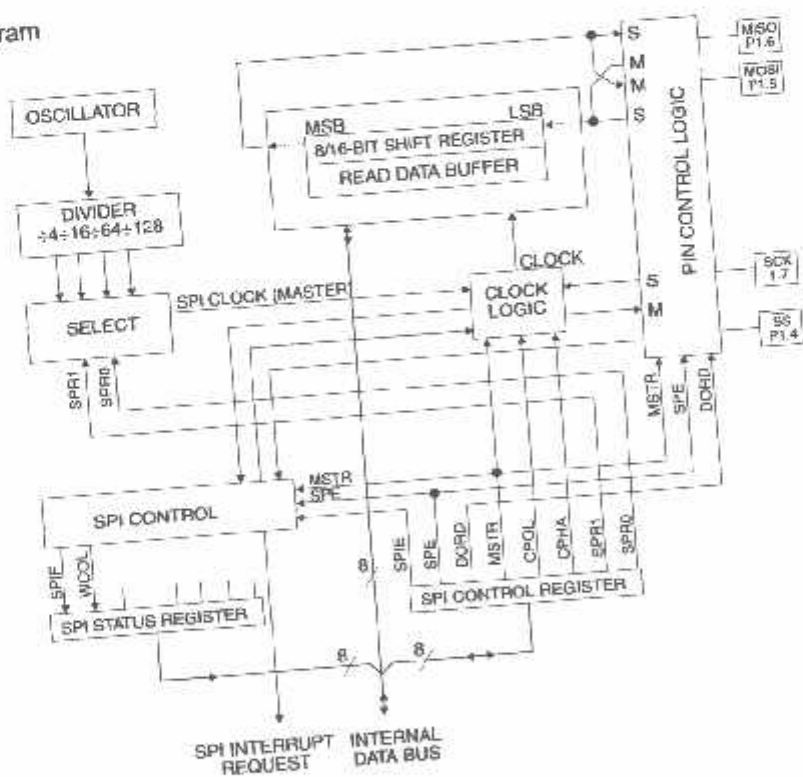
$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (RCAP2H, RCAP2L)]}$$

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

# 5. Timer 2 in Clock-out Mode



## i. SPI Block Diagram



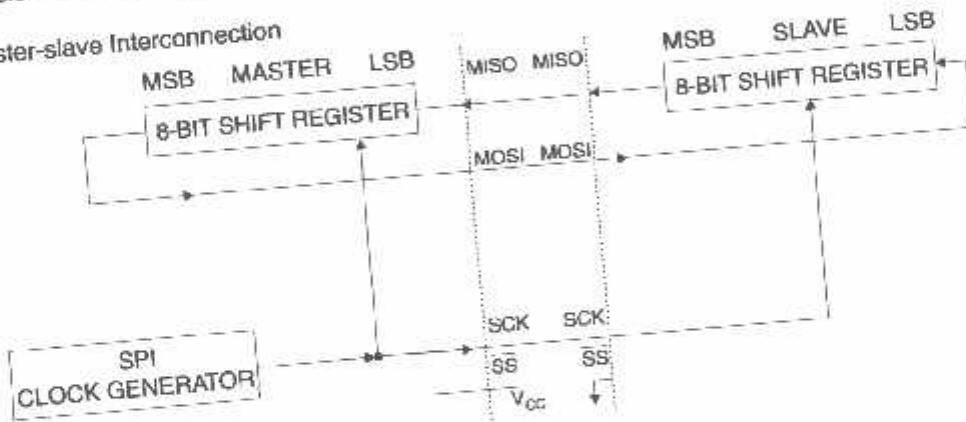
UART in the AT89S8252 operates the same way as UART in the AT89C51, AT89C52 and AT89C55. For information, see the October 1995 Microcontroller book, page 2-49, section titled, "Serial Interface."

## 1 Peripheral Interface

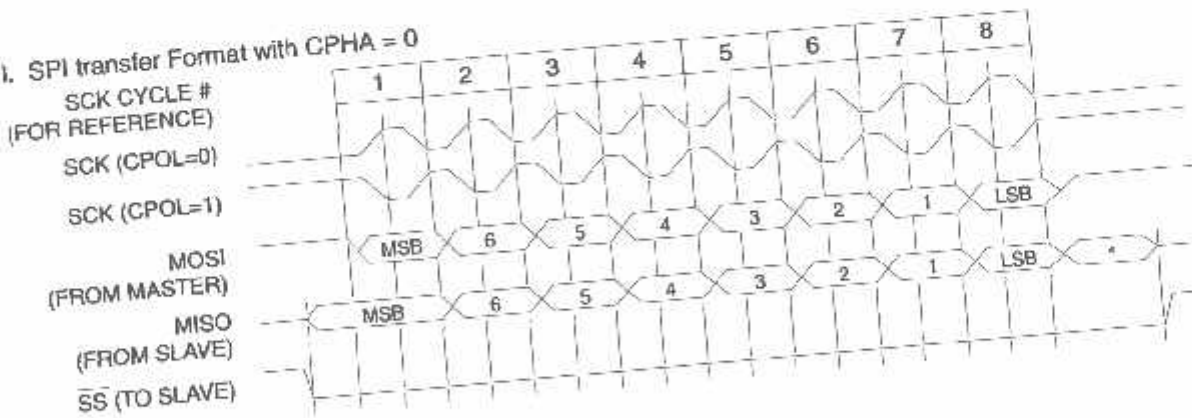
Serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and other serial devices or between several AT89S8252s. The AT89S8252 SPI features include the following:

- Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 10 MHz Bit Frequency (max.)
- First or MSB First Data Transfer
- Programmable Bit Rates
- SPI Transmission Interrupt Flag

### 7. SPI Master-slave Interconnection



### 1. SPI transfer Format with CPHA = 0



... MSB of character just received

- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

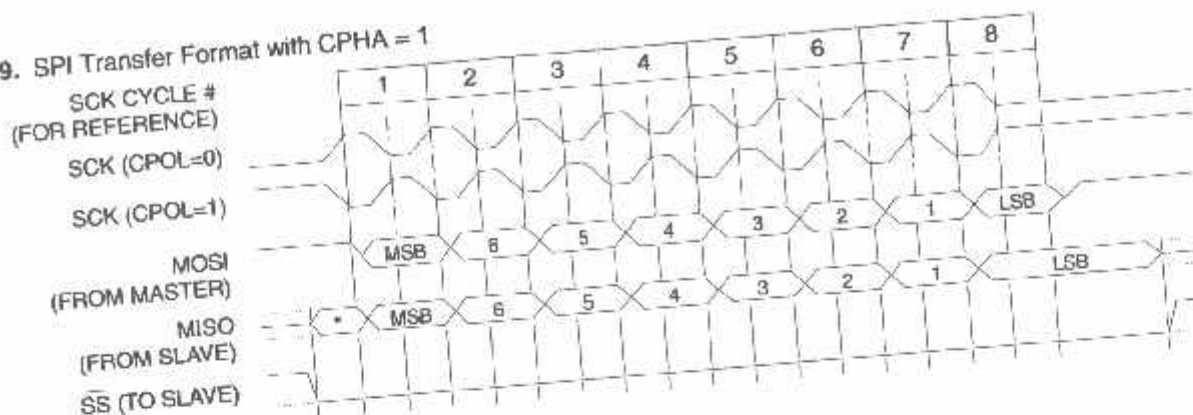
The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MOSI pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested.

The Slave Select input,  $\overline{SS}/P1.4$ , is set low to select an individual SPI device as a slave. When  $\overline{SS}/P1.4$  is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 8 and Figure 9.



## 9. SPI Transfer Format with CPHA = 1



finished but normally LSB of previously transmitted character

## Interrupts

AT89S8252 has a total of six interrupt vectors: two timer interrupts (INT0 and INT1), three timer interrupts (0, 1, and 2), and the serial port interrupt. These are all shown in Figure 10.

These interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bits, since they may be used in future AT89

interrupt is generated by the logical OR of bits TF2 and TF1 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to the interrupt. The service routine may have to determine if it was TF2 or EXF2 that generated the interrupt, and the bit will have to be cleared in software.

Timer 0 and Timer 1 flags, TF0 and TF1, are set at the end of the cycle in which the timers overflow. The values are polled by the circuitry in the next cycle. However, timer 2 flag, TF2, is set at S2P2 and is polled in the cycle in which the timer overflows.

Table 10. Interrupt Enable (IE) Register

(MSB)(LSB)

EA    —    ET2   ES   ET1   EX1   ET0   EX0

Enable Bit = 1 enables the interrupt.

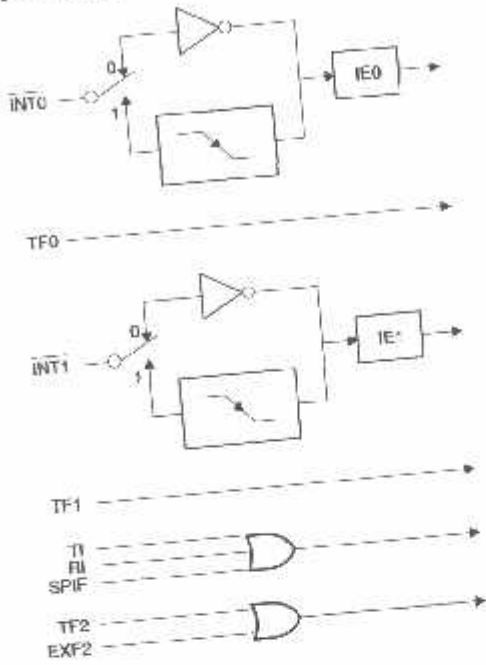
Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	SPI and UART interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.



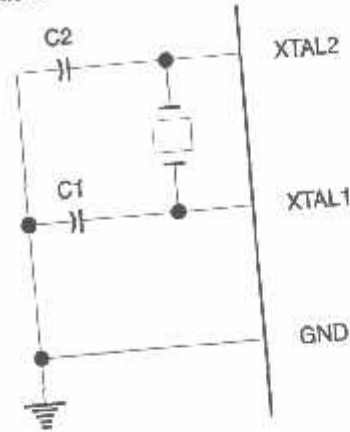
## 10. Interrupt Sources



## Oscillator Characteristics

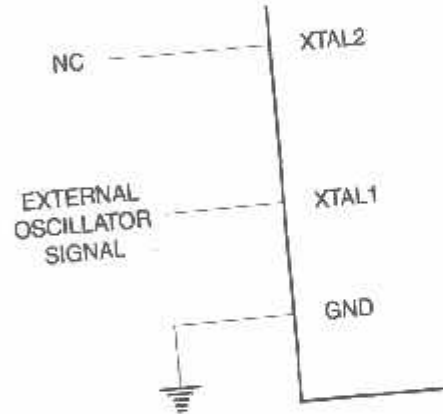
XTAL1 and XTAL2 are the input and output, respectively, of an on-chip inverting amplifier that can be configured for use as an oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the oscillator from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external signal, since the input to the internal clocking circuitry is a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 11. Oscillator Connections



Note: Note: C1, C2 = 30 pF ± 10 pF for Crystals  
= 40 pF ± 10 pF for Ceramic Resonators

Figure 12. External Clock Drive Configuration







## Mode

mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by the instruction that invokes power-down. The content of the on-chip RAM and all the special function registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

## States of External Pins During Idle and Power-down Modes

	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
	Internal	1	1	Data	Data	Data	Data
	External	1	1	Float	Data	Address	Data
down	Internal	0	0	Data	Data	Data	Data
down	External	0	0	Float	Data	Data	Data

## Power-down Mode

Power-down mode, the oscillator is stopped and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is exited. Exit from power-down can be initiated either by an interrupt or by an enabled external interrupt. Reset is not reset or by an enabled external interrupt. Reset resets the SFRs but does not change the on-chip RAM. Power-down should not be activated before  $V_{CC}$  is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Power-down via an interrupt, the external interrupt is enabled as level sensitive before entering power-down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

## Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

When lock bit 1 is programmed, the logic level at the  $\overline{EA}$  pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of  $\overline{EA}$  must agree with the current logic level at that pin in order for the device to function properly.

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

## Bit Protection Modes<sup>(1)(2)</sup>

Program Lock Bits			Protection Type
LB1	LB2	LB3	
U	U	U	No internal memory lock feature.
P	U	U	MOVX instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{EA}$ is sampled and latched on reset and further programming of the Flash memory (parallel or serial mode) is disabled.
P	P	U	Same as Mode 2, but parallel or serial verify are also disabled.
P	P	P	Same as Mode 3, but external execution is also disabled.

1. U = Unprogrammed  
2. P = Programmed

AT89S8252

## Programming the Flash and EEPROM

The AT89S8252 Flash Microcontroller offers 8K bytes of system reprogrammable Flash Code memory and 2K of EEPROM Data memory.

The AT89S8252 is normally shipped with the on-chip Flash and EEPROM Data memory arrays in the erased state (i.e., contents = FFH) and ready to be programmed. The device supports a High-voltage (12V) Parallel programming mode and a Low-voltage (5V) Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 into a host's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In parallel programming mode, the two arrays occupy one continuous address space: 0000H to 1FFFFH for the Code array and 2000H to 27FFFH for the Data array.

Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode as long as any of the lock bits have been programmed.

In parallel programming mode, there is no auto-erase cycle. To reprogram any non-blank byte, the user needs to perform the Chip Erase operation first to erase both arrays.

**Programming Algorithm:** To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

1. Power-up sequence:

Apply power between  $V_{CC}$  and GND pins.

2. RST pin to "H".

3. Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait at least 10 milliseconds.

4.  $\overline{PSEN}$  pin to "L".

5.  $\overline{PSEN}$  pin to "H".

6. Set  $\overline{EA}$  pin to "H" and all other pins to "H".

7. Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.

8. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.

9. Apply data to pins P0.0 to P0.7 for Write Code operation.

5. Raise  $\overline{EA}/V_{PP}$  to 12V to enable Flash programming, erase or verification.
6. Pulse  $\overline{ALE}/\overline{PROG}$  once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
7. To verify the byte just programmed, bring pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.
8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:
  - Set XTAL1 to "L".
  - Set RST and  $\overline{EA}$  pins to "L".
  - Turn  $V_{CC}$  power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

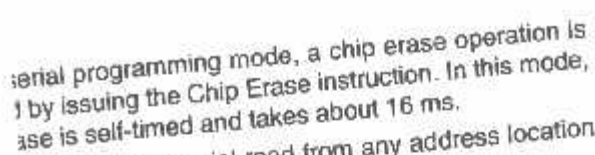
**Data Polling:** The AT89S8252 features  $\overline{DATA}$  Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin.  $\overline{DATA}$  Polling may begin any time after a write cycle has been initiated.

**Ready/Busy:** The progress of byte programming in the parallel programming mode can also be monitored by the  $\overline{RDY}/\overline{BSY}$  output signal. Pin P3.4 is pulled Low after  $\overline{ALE}$  goes High during programming to indicate  $\overline{BSY}$ . P3.4 is pulled High again when programming is done to indicate  $\overline{RDY}$ .

**Program Verify:** If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

**Chip Erase:** Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding  $\overline{ALE}/\overline{PROG}$  low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation.





chip erase, a serial read from any address location (00H at the data outputs).

**Programming Fuse:** A programmable fuse is available for disabling Serial Programming if the user needs an system security. The Serial Programming Fuse may be programmed or erased in the Parallel Programming mode.

89S8252 is shipped with the Serial Programming enabled. The signature bytes are

**g the Signature Bytes:** The signature bytes are the same procedure as a normal verification of 030H and 031H, except that P3.6 and P3.7 must be tied to a logic low. The values returned are as follows:

- 030H) = 1EH indicates manufactured by Atmel
- 031H) = 72H indicates 89S8252

## ramming Interface

code byte in the Flash and EEPROM arrays can be erased, by using the appropriate combination of control signals. The write cycle is self-timed and once initiated, will automatically time itself to completion.

For programming vendors offer worldwide support for all microcontroller series. Please contact your local programming vendor for the appropriate software revision:

Downloading

**Downloading**

Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to serial interface consists of pins SCK, MOSI (input/output). After RST is set high, the Programming instruction needs to be executed first before programming operations can be executed.

erase cycle is built into the self-timed programming (in the serial mode ONLY) and there is no need to execute the Chip Erase instruction unless any of the bits have been programmed. The Chip Erase operation erases the content of every memory location in both the Data and Data arrays into FFH.

Instruction and Data memory arrays have separate address

0000H to 1FFFH for Code memory and 000H to 7FFH for Data memory.

0000H to 1FFFH.  
Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.

## Serial Programming Algorithm

**Serial Programming Algorithm**

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:  
Apply power between VCC and GND pins.  
Set RST pin to "H".  
If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
  2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.
  3. The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.
  4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
  5. At the end of a programming session, RST can be set low to commence normal operation.
- Power-off sequence (if needed):  
Set XTAL1 to "L" (if a crystal is not used).  
Set RST to "L".  
Turn  $V_{CC}$  power off.

## Serial Programming Instruction

### Serial Programming Instruction

The Instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:

## Instruction Set

Instruction	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high.
Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 2K memory arrays.
Code Memory	aaaa a001	low addr	xxxx xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Code Memory	aaaa a010	low addr	data in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Data Memory	00aa a101	low addr	xxxx xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Data Memory	00aa a110	low addr	data in	Write data to Data memory location at selected address.
Lock Bits	1010 1100	x x111	xxxx xxxx	Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits.

1. DATA polling is used to indicate the end of a write cycle which typically takes less than 2.5 ms at 5V.
2. "aaaaa" = high order address.
3. "x" = don't care.



# 1 and EEPROM Parallel Programming Modes

	RST	PSEN	ALE/PROG	EA/V <sub>PP</sub>	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Prog. Modes	H	H <sup>(1)</sup>	H <sup>(1)</sup>	x					X	X
Erase	H	L		12V	H	L	L	L	DIN	ADDR
0K bytes) Memory	H	L		12V	L	H	H	H	DOUT	ADDR
0K bytes) Memory	H	L	H	12V	L	L	H	H	DIN	X
Lock Bits:	H	L		12V	H	L	H	L	P0.7 = 0	X
									P0.6 = 0	X
									P0.5 = 0	X
Bit - 1									DOUT	X
Bit - 2									@P0.2	X
Bit - 3									@P0.1	X
	H	L	H	12V	H	H	L	L	@P0.0	X
Bit - 1									DOUT	30H
Bit - 2									DOUT	31H
Bit - 3										
Intel Code	H	L	H	12V	L	L	L	L	P0.0 = 0	X
Device Code	H	L		12V	L	H	L	H	P0.0 = 1	X
Prog. Enable	H	L		12V	L	H	L	H	@P0.0	X
Prog. Disable	H	L	H	12V	H	H	L	H		
Serial Prog. Fuse	H	L								

1. "h" = weakly pulled "High" internally.

2. Chip Erase and Serial Programming Fuse require a 10 ms PROG pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.

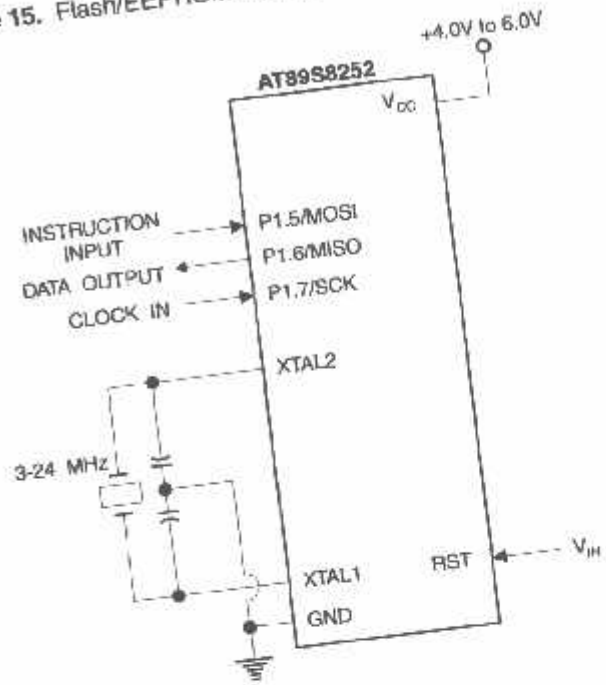
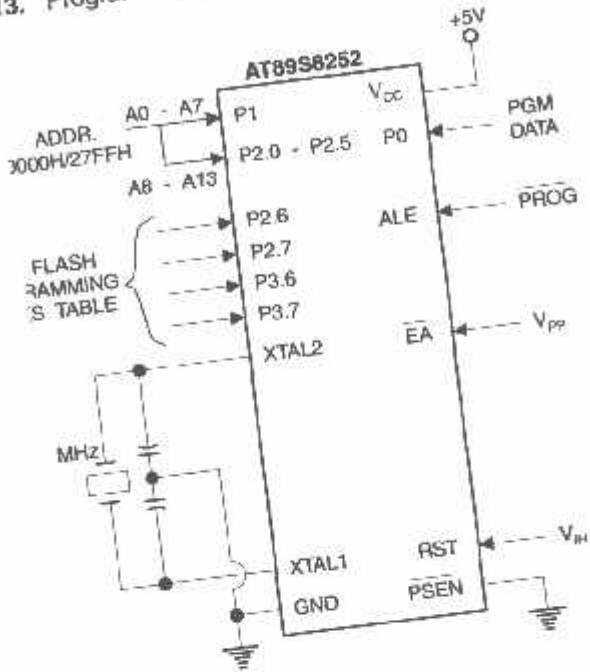
3. P3.4 is pulled Low during programming to indicate RDY/BSY.

4. "X" = don't care

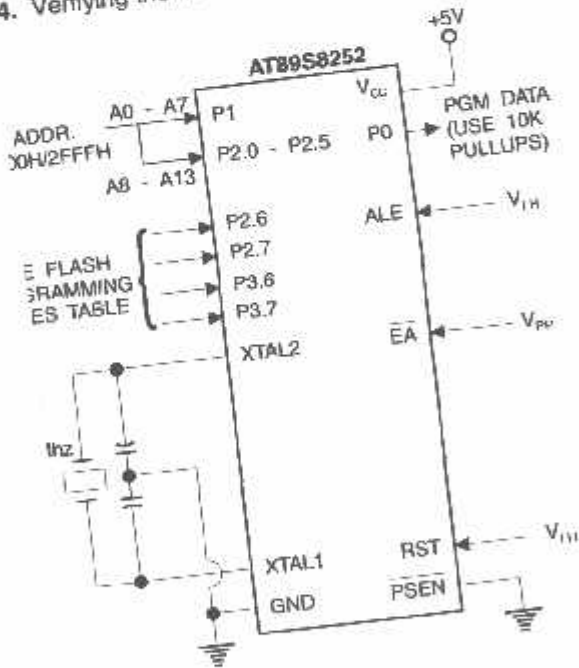
AT89S8252

## 13. Programming the Flash/EEPROM Memory

Figure 15. Flash/EEPROM Serial Downloading



## 14. Verifying the Flash/EEPROM Memory







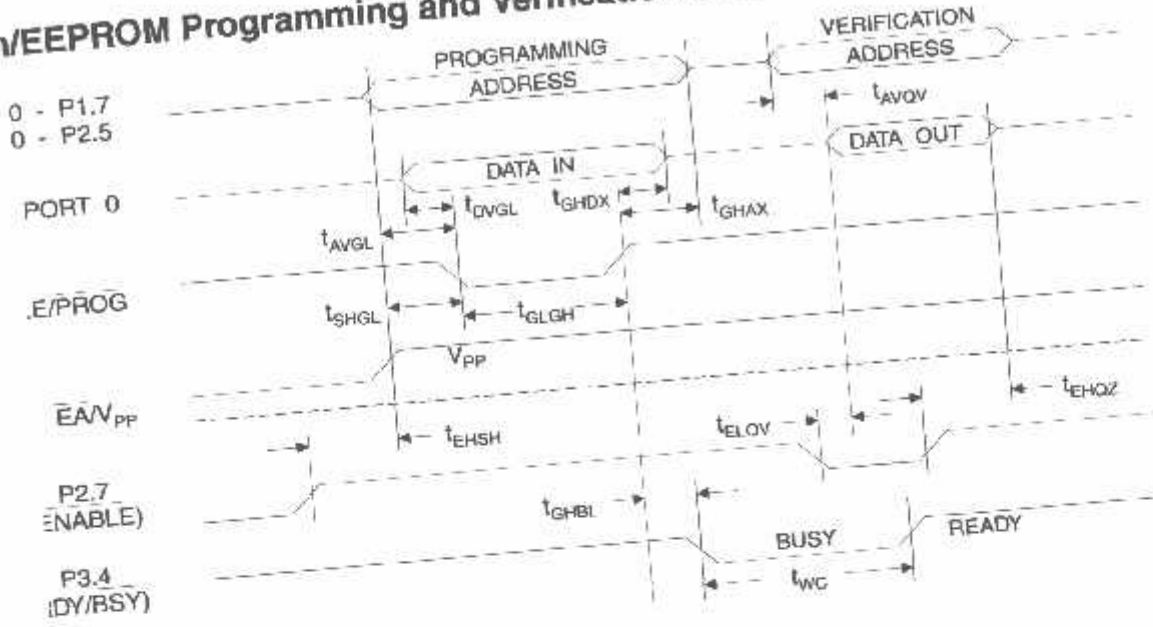
## Programming and Verification Characteristics – Parallel Mode

to 70°C,  $V_{CC} = 5.0V \pm 10\%$

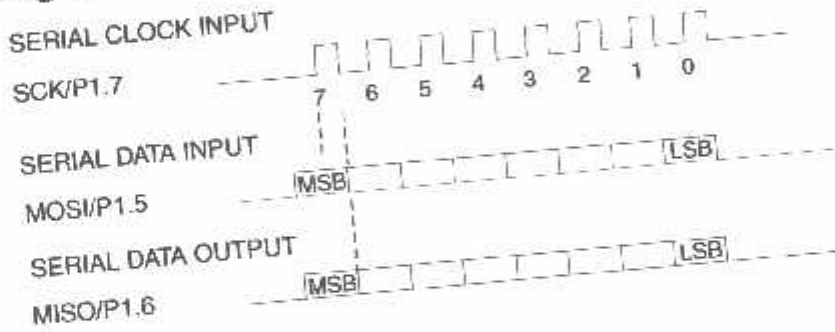
Parameter	Min	Max	Units
Programming Enable Voltage	11.5	12.5	V
Programming Enable Current		1.0	mA
Oscillator Frequency	3	24	MHz
Address Setup to PROG Low	$48t_{CLCL}$		
Address Hold after PROG	$48t_{CLCL}$		
Data Setup to PROG Low	$48t_{CLCL}$		
Data Hold after PROG	$48t_{CLCL}$		
P2.7 (ENABLE) High to $V_{PP}$	10		$\mu s$
$V_{PP}$ Setup to PROG Low	1	110	$\mu s$
PROG Width		$48t_{CLCL}$	
Address to Data Valid		$48t_{CLCL}$	
ENABLE Low to Data Valid	0	$48t_{CLCL}$	
Data Float after ENABLE		1.0	$\mu s$
PROG High to BUSY Low		2.0	ms
Byte Write Cycle Time			

AT89S8252

# EEPROM Programming and Verification Waveforms – Parallel Mode



## I Downloading Waveforms







## Absolute Maximum Ratings\*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
Input Current	15.0 mA

### \*NOTICE:

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## Characteristics

Values shown in this table are valid for  $T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$  and  $V_{CC} = 5.0\text{V} \pm 20\%$ , unless otherwise noted.

Parameter	Condition	Min	Max	Units
Input Low-voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
Input Low-voltage (EA)		-0.5	$0.2 V_{CC} - 0.3$	V
Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
Input High-voltage (XTAL1, RST)		$0.7 V_{CC}$	$V_{CC} + 0.5$	V
Output Low-voltage <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.5	V
Output Low-voltage <sup>(1)</sup> (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.5	V
Output High-voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
	$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
	$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
Output High-voltage (Port 0 in External Bus Mode)	$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
	$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 2\text{V}, V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$
Logical 1 to 0 Transition Current (Ports 1,2,3)	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
Input Leakage Current (Port 0, EA)		50	300	K $\Omega$
Reset Pull-down Resistor	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
Pin Capacitance	Active Mode, 12 MHz		25	mA
Power Supply Current	Idle Mode, 12 MHz		6.5	mA
	$V_{CC} = 6\text{V}$		100	$\mu\text{A}$
Power-down Mode <sup>(2)</sup>	$V_{CC} = 3\text{V}$		40	$\mu\text{A}$

Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per port pin: 10 mA  
Maximum  $I_{OL}$  per 8-bit port:  
Port 0: 26 mA  
Ports 1, 2, 3: 15 mA

Maximum total  $I_{OL}$  for all output pins: 71 mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.  
2. Minimum  $V_{CC}$  for Power-down is 2V

# AT89S8252

## Characteristics

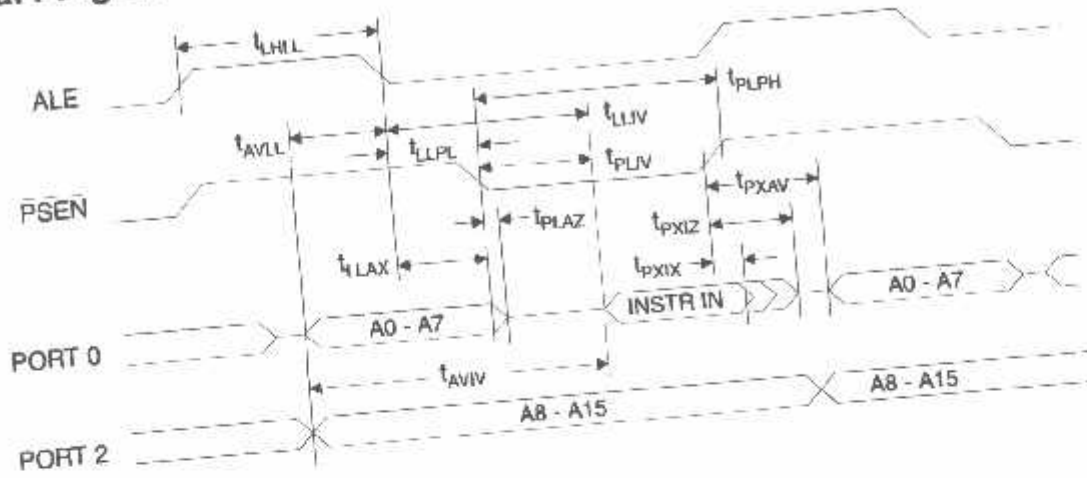
Operating conditions, load capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; load capacitance for all other = 80 pF.

### Normal Program and Data Memory Characteristics

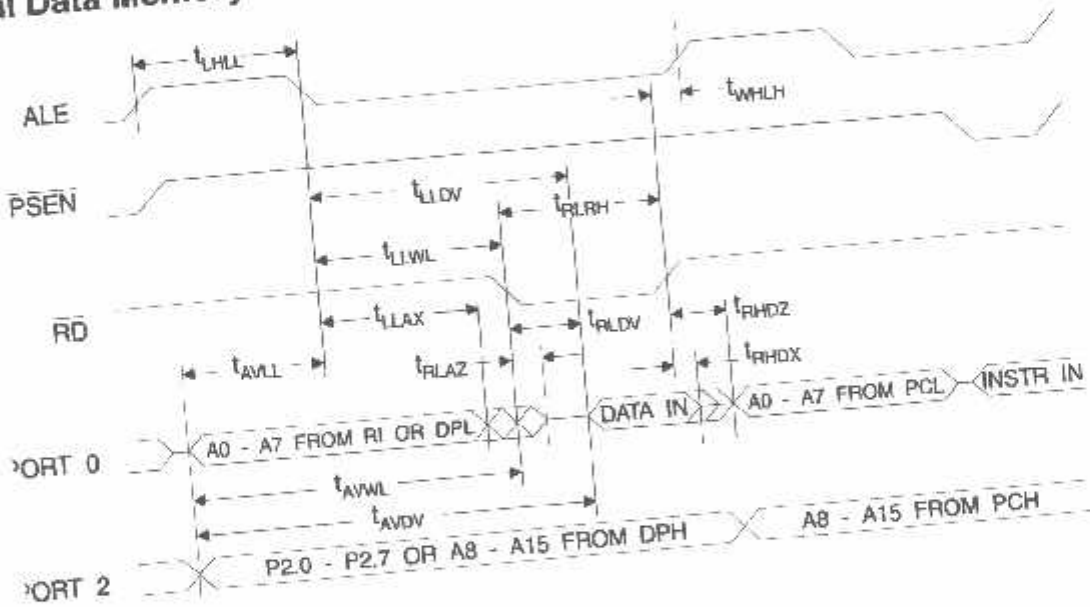
Parameter	Variable Oscillator		Units
	Min	Max	
Oscillator Frequency	0	24	MHz
ALE Pulse Width	$2t_{CLCL} - 40$		ns
Address Valid to ALE Low	$t_{CLCL} - 13$		ns
Address Hold after ALE Low	$t_{CLCL} - 20$		ns
ALE Low to Valid Instruction In		$4t_{CLCL} - 65$	ns
ALE Low to PSEN Low	$t_{CLCL} - 13$		ns
PSEN Pulse Width	$3t_{CLCL} - 20$		ns
PSEN Low to Valid Instruction In		$3t_{CLCL} - 45$	ns
Input Instruction Hold after PSEN	0	$t_{CLCL} - 10$	ns
Input Instruction Float after PSEN	$t_{CLCL} - 8$		ns
PSEN to Address Valid		$5t_{CLCL} - 55$	ns
Address to Valid Instruction In		10	ns
PSEN Low to Address Float	$6t_{CLCL} - 100$		ns
RD Pulse Width	$6t_{CLCL} - 100$		ns
WR Pulse Width		$5t_{CLCL} - 90$	ns
RD Low to Valid Data In	0		ns
Data Hold after RD		$2t_{CLCL} - 28$	ns
Data Float after RD		$8t_{CLCL} - 150$	ns
ALE Low to Valid Data In		$9t_{CLCL} - 165$	ns
Address to Valid Data In	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
ALE Low to RD or WR Low	$4t_{CLCL} - 75$		ns
Address to RD or WR Low	$t_{CLCL} - 20$		ns
Data Valid to WR Transition	$7t_{CLCL} - 120$		ns
Data Valid to WR High	$t_{CLCL} - 20$		ns
Data Hold after WR		0	ns
RD Low to Address Float	$t_{CLCL} - 20$	$t_{CLCL} + 25$	ns
RD or WR High to ALE High			



## Instruction Program Memory Read Cycle

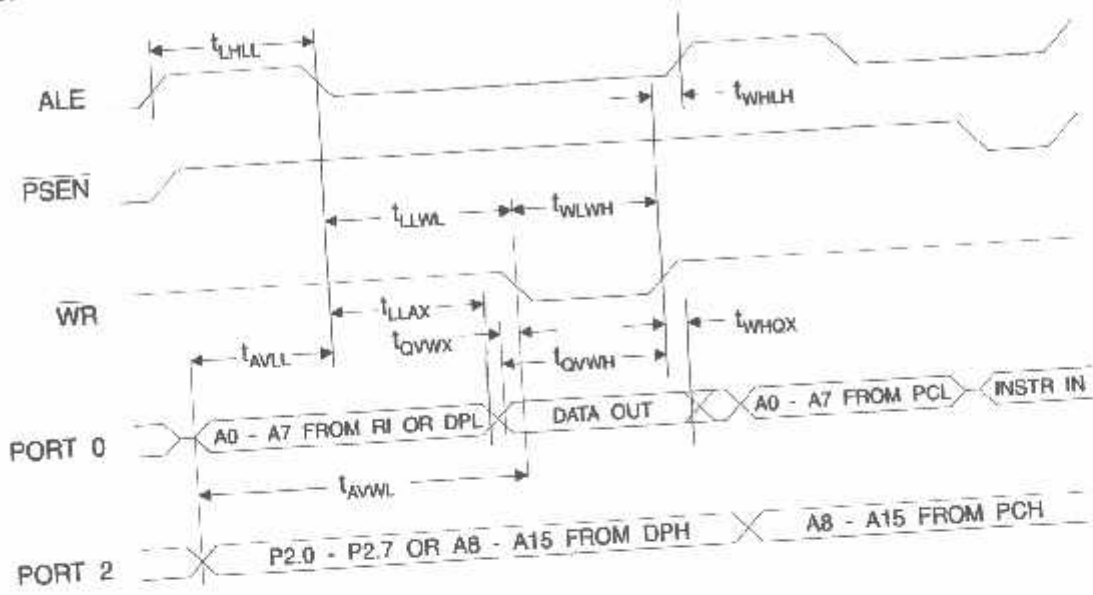


## Instruction Data Memory Read Cycle

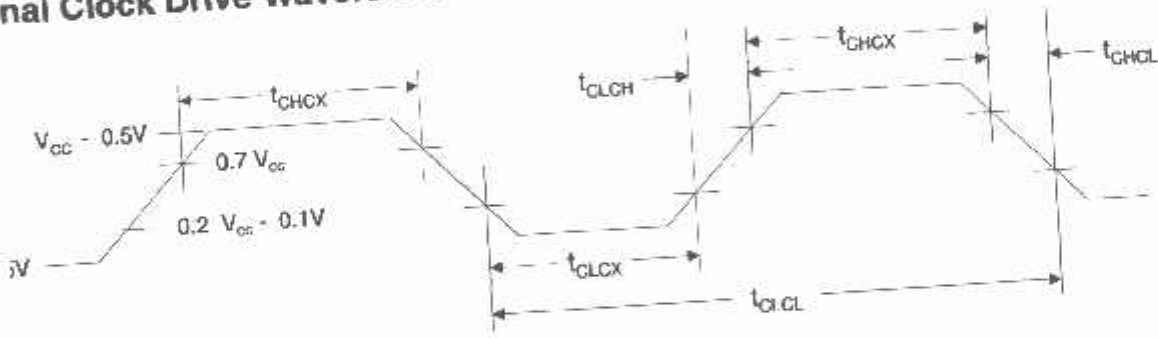


AT89S8252

### nal Data Memory Write Cycle



### nal Clock Drive Waveforms



### nal Clock Drive

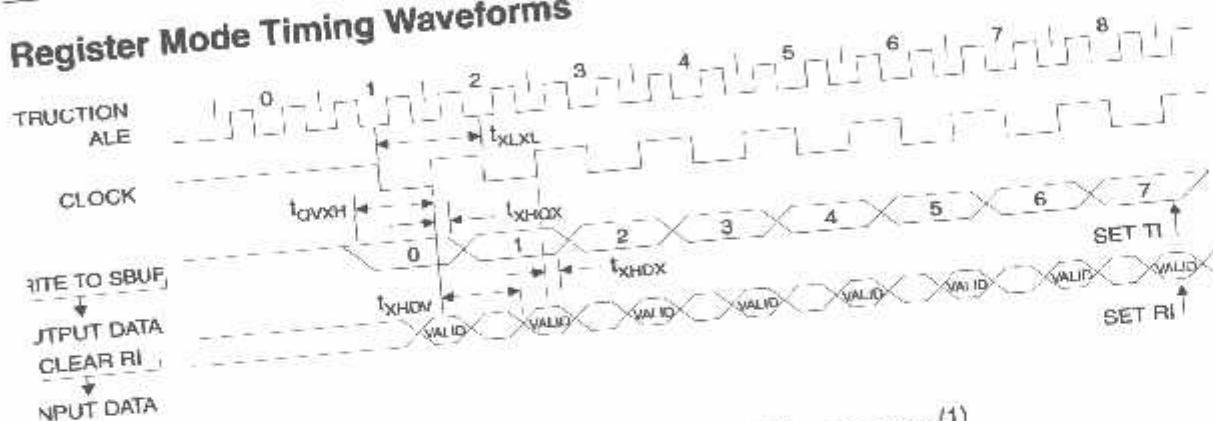
Parameter	$V_{CC} = 4.0V$ to $6.0V$		Units
	Min	Max	
Oscillator Frequency	0	24	MHz
Clock Period	41.6		ns
High Time	15		ns
Low Time		20	ns
Rise Time		20	ns
Fall Time			ns



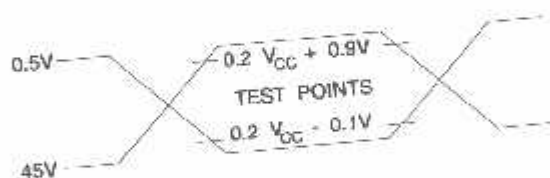
**Timing: Shift Register Mode Test Conditions**  
 Values in this table are valid for  $V_{CC} = 4.0V$  to  $6V$  and Load Capacitance =  $80\text{ pF}$ .

Timing: Shift Register Mode Test Conditions		Variable Oscillator		Units
Parameter		Min	Max	
Serial Port Clock Cycle Time		$12t_{CLCL}$		$\mu s$
Output Data Setup to Clock Rising Edge		$10t_{CLCL} - 133$		ns
Output Data Hold after Clock Rising Edge		$2t_{CLCL} - 117$		ns
Input Data Hold after Clock Rising Edge		0		ns
Clock Rising Edge to Input Data Valid			$10t_{CLCL} - 133$	ns

### Register Mode Timing Waveforms

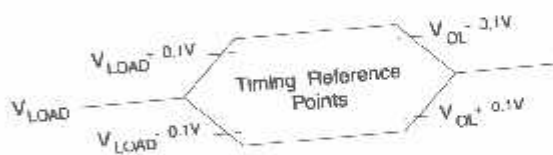


### Testing Input/Output Waveforms<sup>(1)</sup>



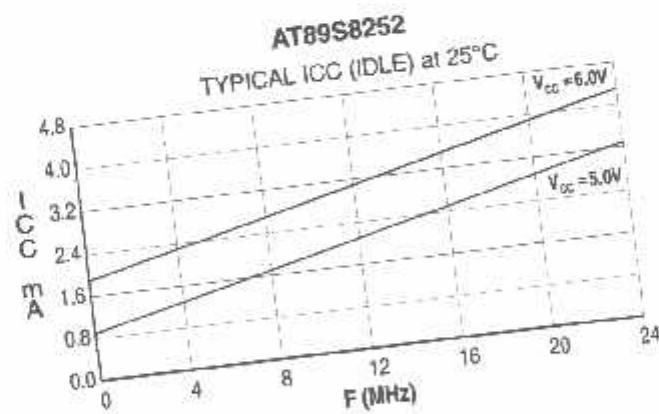
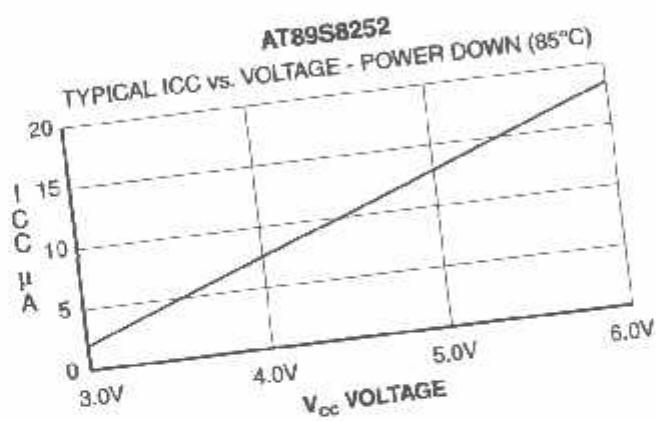
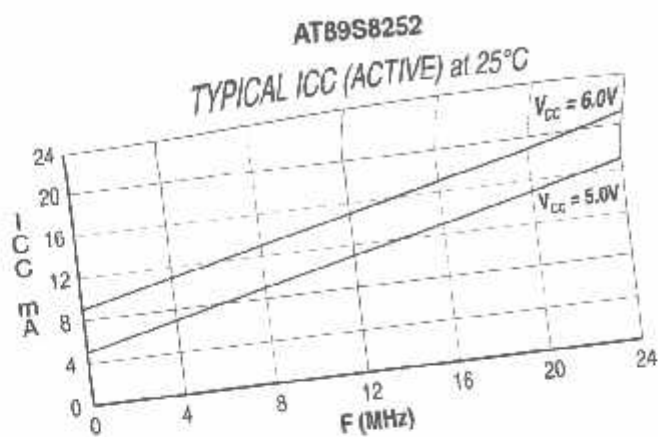
1. AC Inputs during testing are driven at  $V_{CC} - 0.5V$  for a logic 1 and  $0.45V$  for a logic 0. Timing measurements are made at  $V_{IH}$  min. for a logic 1 and  $V_{IL}$  max. for a logic 0.

### Float Waveforms<sup>(1)</sup>



Notes: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded  $V_{OH}/V_{OL}$  level occurs.

**AT89S8252**



- Notes:
1. XTAL1 tied to GND for  $I_{CC}$  (power-down)
  2. Lock bits programmed



## 1g Information

Power Supply	Ordering Code	Package	Operation Range
4.0V to 6.0V	AT89S8252-24AC	44A	Commercial (0°C to 70°C)
	AT89S8252-24JC	44J	
	AT89S8252-24PC	40P6	
	AT89S8252-24QC	44Q	
4.0V to 6.0V	AT89S8252-24AJ	44A	Industrial (-40°C to 85°C)
	AT89S8252-24JI	44J	
	AT89S8252-24PI	40P6	
	AT89S8252-24QI	44Q	
4.5V to 5.5V	AT89S8252-33AC	44A	Commercial (0°C to 70°C)
	AT89S8252-33JC	44J	
	AT89S8252-33PC	40P6	
	AT89S8252-33QC	44Q	

= Preliminary Information

Package Type
44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44-lead, Plastic J-leaded Chip Carrier (PLCC)
40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

## AT89S8252







## Headquarters

**Atmel Headquarters**  
Orchard Parkway  
Fremont, CA 94531  
(408) 441-0311  
(408) 487-2600

Atmel U.K., Ltd.  
Atmel Business Centre  
The Valley Way  
Surrey, Surrey GU15 3YL  
UK  
(44) 1276-686-677  
(44) 1276-686-697

Atmel Asia, Ltd.  
Room 1219  
The Richem Golden Plaza  
Jody Road Tsimshatsui  
Kowloon  
Hong Kong  
(852) 2721-9778  
(852) 2722-1369

Atmel Japan K.K.  
Tonetsu Shinkawa Bldg.  
1-8 Shinkawa  
Chiyoda-ku, Tokyo 104-0033  
Japan  
(81) 3-3523-3551  
(81) 3-3523-7581

## Atmel Operations

**Atmel Colorado Springs**  
1150 E. Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL (719) 576-3300  
FAX (719) 540-1759

**Atmel Rousset**  
Zone Industrielle  
13106 Rousset Cedex  
France  
TEL (33) 4-4253-6000  
FAX (33) 4-4253-6001

---

### *Fax-on-Demand*

North America:  
1-(800) 292-8635  
International:  
1-(408) 441-0732

*e-mail*  
literature@atmel.com

*Web Site*  
<http://www.atmel.com>

*BBS*  
1-(408) 436-4309

**Atmel Corporation 2000.**  
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not intended for use as critical components in life support devices or systems.

Atmel, the Atmel logo and/or "A" are registered trademarks and trademarks of Atmel Corporation.

All other product names in this document may be trademarks of others.



Printed on recycled paper.

0401E-02/00/AM



# ULN2001A-ULN2002A ULN2003A-ULN2004A

## SEVEN DARLINGTON ARRAYS

SEVEN DARLINGTONS PER PACKAGE  
OUTPUT CURRENT 500mA PER DRIVER  
(600mA PEAK)  
OUTPUT VOLTAGE 50V  
INTEGRATED SUPPRESSION DIODES FOR  
INDUCTIVE LOADS  
OUTPUTS CAN BE PARALLELED FOR  
HIGHER CURRENT  
TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS  
INPUTS PINNED OPPOSITE OUTPUTS TO  
SIMPLIFY LAYOUT



DIP16

ORDERING NUMBERS: ULN2001A/2A/3A/4A



SO16

ORDERING NUMBERS: ULN2001D/2D/3D/4D

### DESCRIPTION

The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

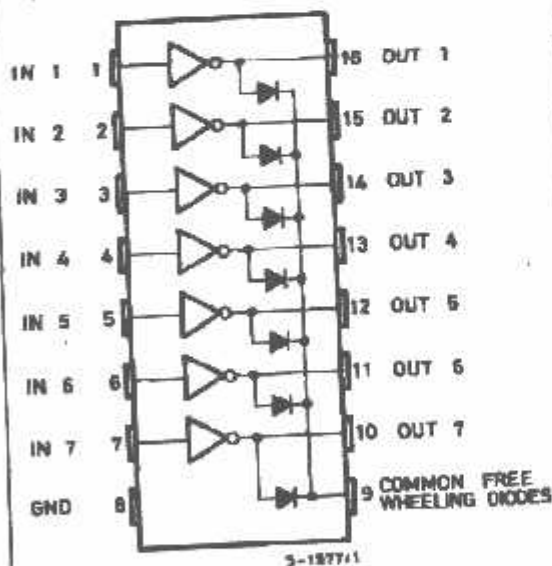
The four versions interface to all common logic families:

ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	6-15V CMOS, PMOS

These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal print-readers and high power buffers.

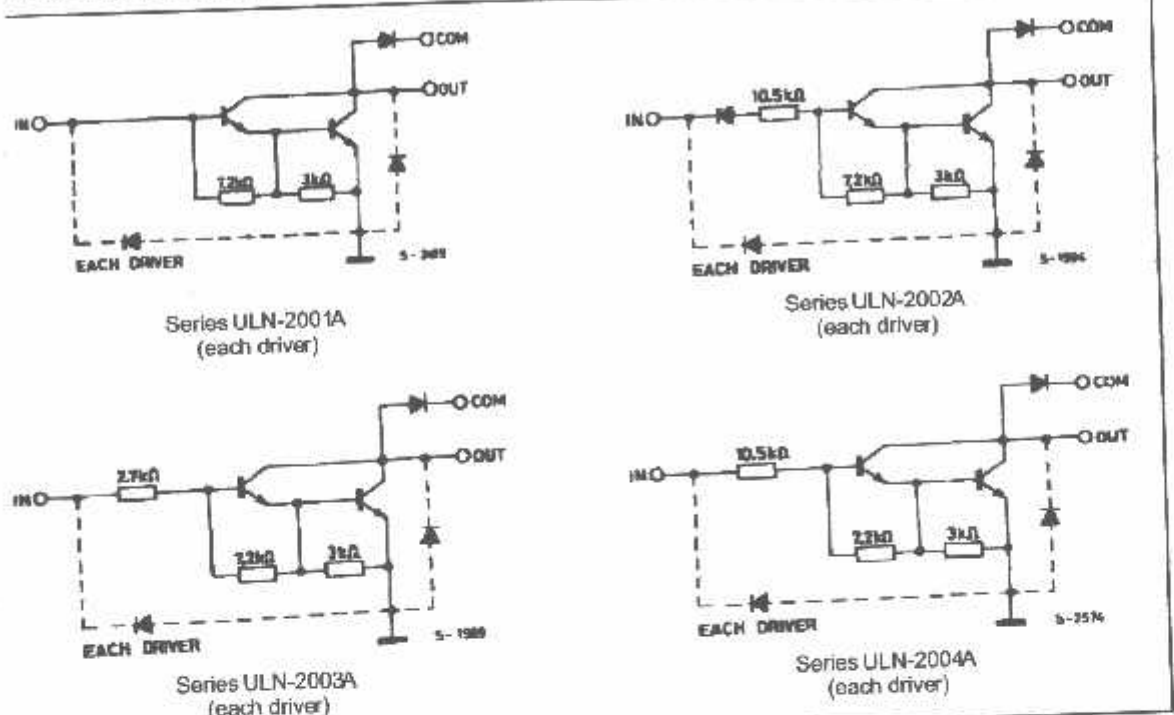
The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper padframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.

### PIN CONNECTION



N2001A - ULN2002A - ULN2003A - ULN2004A

HEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
$V_o$	Output Voltage	50	V
$V_{in}$	Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
$I_c$	Continuous Collector Current	500	mA
$I_b$	Continuous Base Current	25	mA
$T_{amb}$	Operating Ambient Temperature Range	- 20 to 85	°C
$T_{stg}$	Storage Temperature Range	- 55 to 150	°C
$T_j$	Junction Temperature	150	°C

THERMAL DATA

Symbol	Parameter	DIP16	SO16	Unit
$R_{th(j-amb)}$	Thermal Resistance Junction-ambient	70	100	°C/W



# ULN2001A - ULN2002A - ULN2003A - ULN2004A

## ELECTRICAL CHARACTERISTICS ( $T_{amb} = 25^{\circ}\text{C}$ unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
$I_{CEX}$	Output Leakage Current	$V_{CE} = 50\text{V}$ $T_{amb} = 70^{\circ}\text{C}$ , $V_{CE} = 50\text{V}$			50 100	$\mu\text{A}$ $\mu\text{A}$	1a 1a
		$T_{amb} = 70^{\circ}\text{C}$ for ULN2002A $V_{CE} = 50\text{V}$ , $V_i = 6\text{V}$			500	$\mu\text{A}$	1b
		for ULN2004A $V_{CE} = 50\text{V}$ , $V_i = 1\text{V}$			500	$\mu\text{A}$	1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	$I_C = 100\text{mA}$ , $I_B = 250\mu\text{A}$		0.9	1.1	V	2
		$I_C = 200\text{mA}$ , $I_B = 350\mu\text{A}$		1.1	1.3	V	2
		$I_C = 350\text{mA}$ , $I_B = 500\mu\text{A}$		1.3	1.6	V	2
$I_{i(on)}$	Input Current	for ULN2002A, $V_i = 17\text{V}$		0.82	1.25	mA	3
		for ULN2003A, $V_i = 3.85\text{V}$		0.93	1.35	mA	3
		for ULN2004A, $V_i = 5\text{V}$		0.35	0.5	mA	3
		$V_i = 12\text{V}$		1	1.45	mA	3
$I_{i(off)}$	Input Current	$T_{amb} = 70^{\circ}\text{C}$ , $I_C = 500\mu\text{A}$	50	65		$\mu\text{A}$	4
$V_{i(on)}$	Input Voltage	$V_{CE} = 2\text{V}$ for ULN2002A $I_C = 300\text{mA}$ for ULN2003A $I_C = 200\text{mA}$ $I_C = 250\text{mA}$ $I_C = 300\text{mA}$ for ULN2004A $I_C = 125\text{mA}$ $I_C = 200\text{mA}$ $I_C = 275\text{mA}$ $I_C = 350\text{mA}$			13 2.4 2.7 3 5 6 7 8	V	5
$h_{FE}$	DC Forward Current Gain	for ULN2001A $V_{CE} = 2\text{V}$ , $I_C = 350\text{mA}$	1000				2
$C_i$	Input Capacitance			15	25	pF	
$t_{PLH}$	Turn-on Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	$\mu\text{s}$	
$t_{PHL}$	Turn-off Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	$\mu\text{s}$	
$I_R$	Clamp Diode Leakage Current	$V_R = 50\text{V}$ $T_{amb} = 70^{\circ}\text{C}$ , $V_R = 50\text{V}$			50 100	$\mu\text{A}$ $\mu\text{A}$	6 6
				1.7	2	V	7
$V_F$	Clamp Diode Forward Voltage	$I_F = 350\text{mA}$					

### 3T CIRCUITS

Figure 1a.

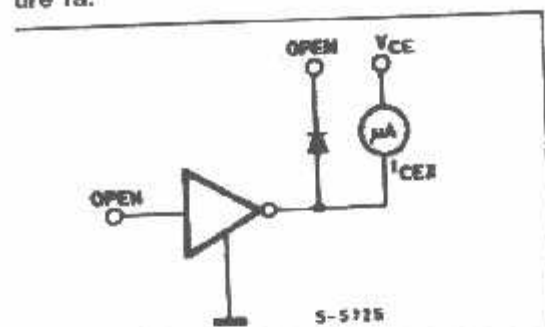


Figure 1b.

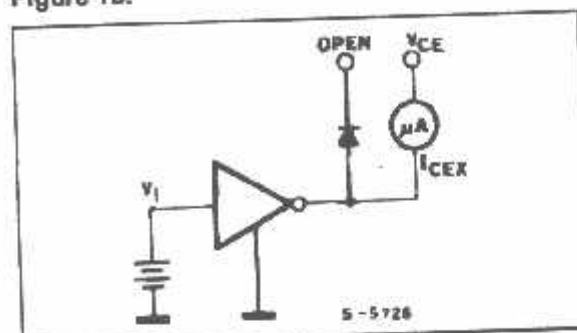


Figure 2.

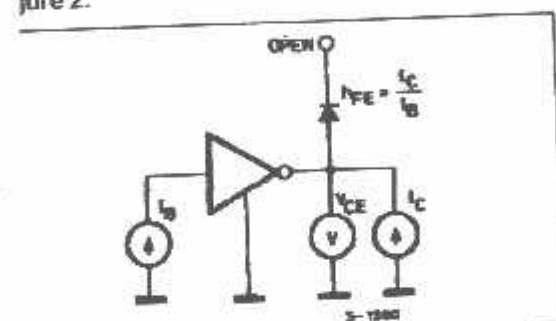


Figure 3.

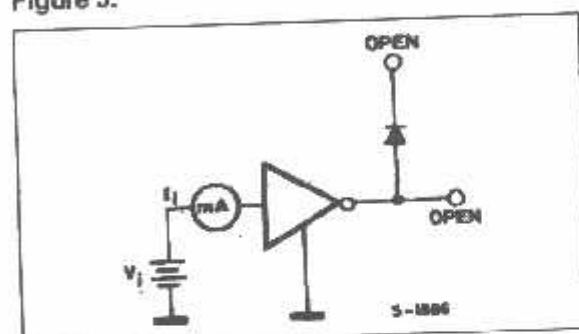


Figure 4.

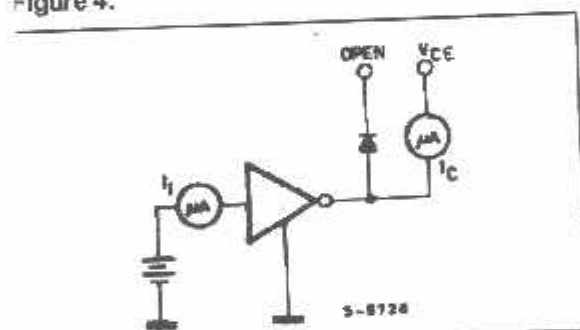


Figure 5.

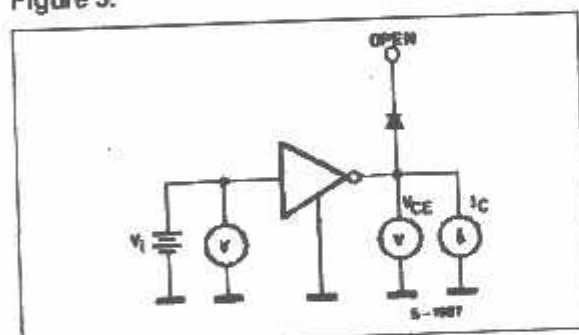


Figure 6.

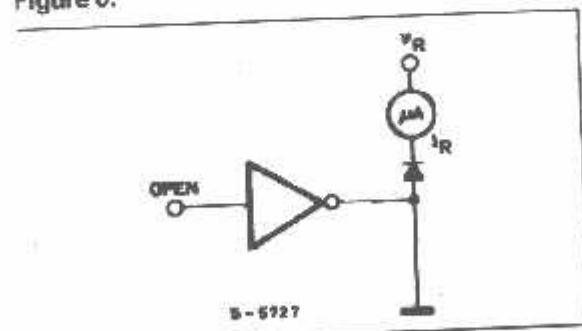


Figure 7.

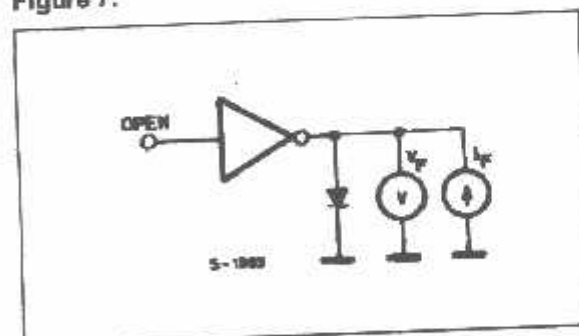


Figure 8: Collector Current versus Input Current

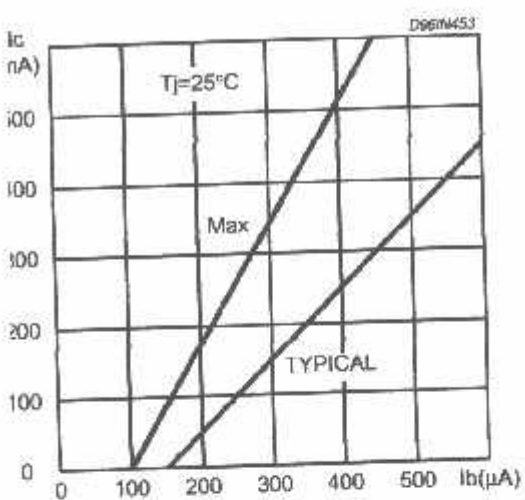


Figure 9: Collector Current versus Saturation Voltage

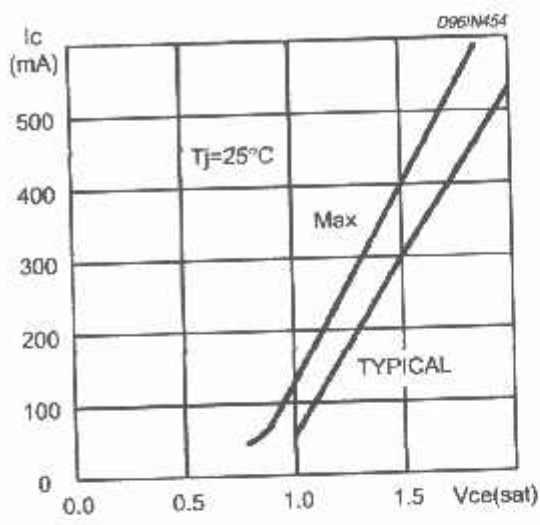


Figure 10: Peak Collector Current versus Duty Cycle

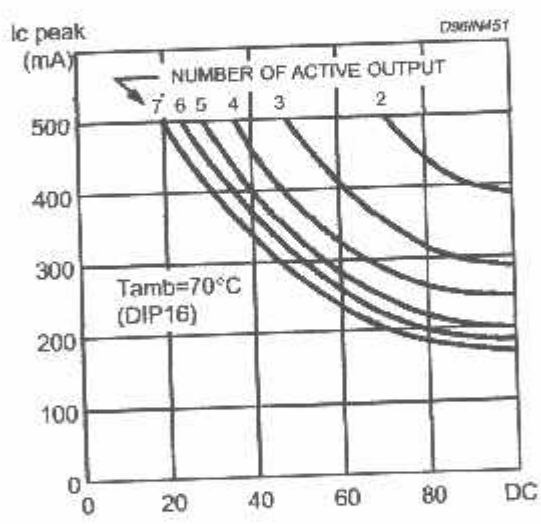
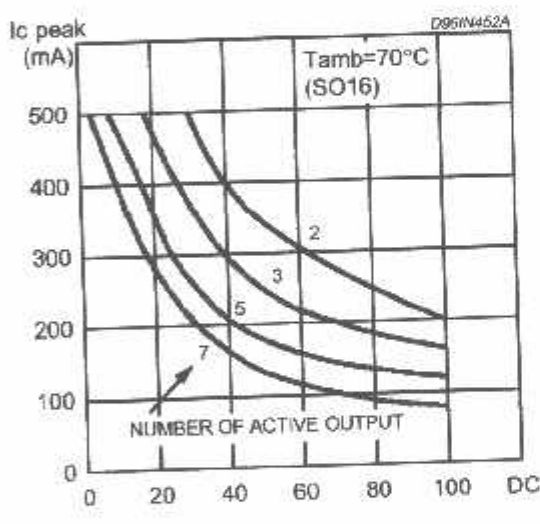
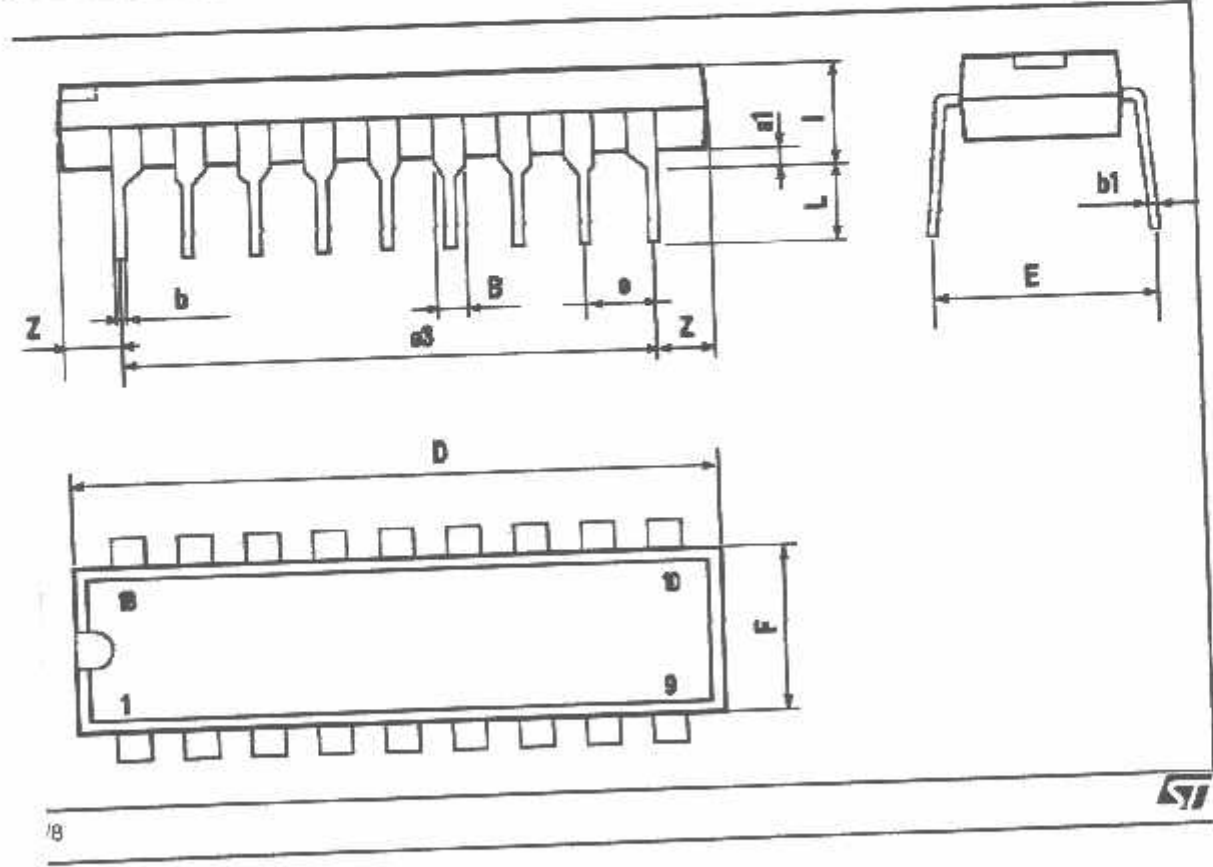


Figure 11: Peak Collector Current versus Duty Cycle



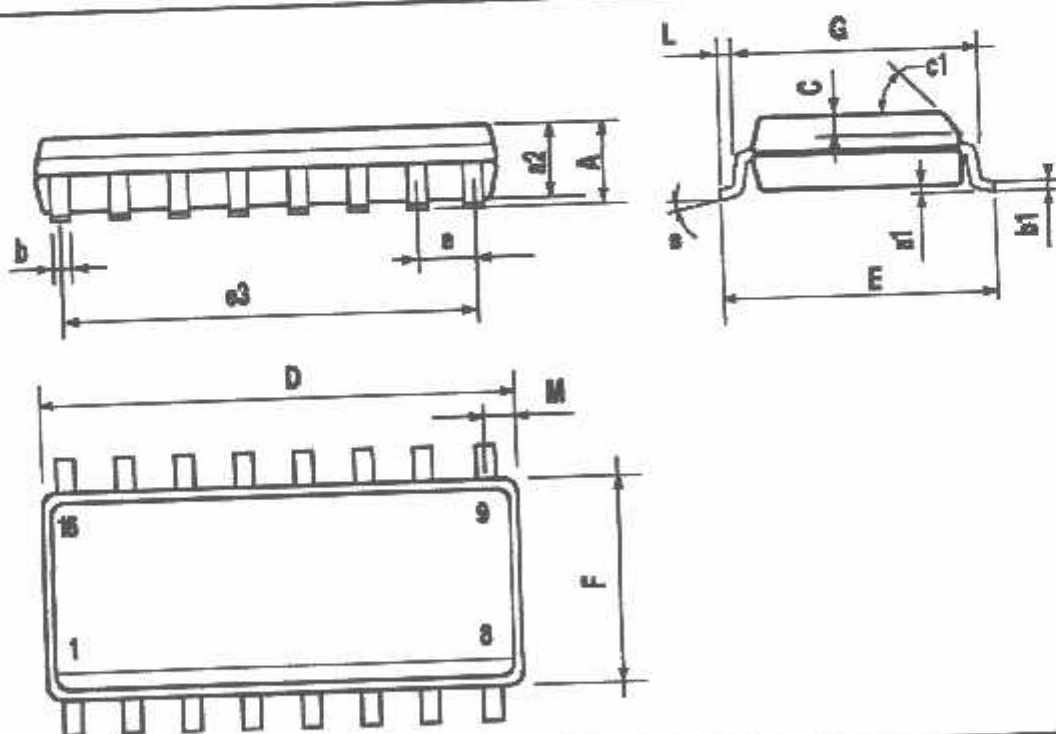
16 PACKAGE MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.51			0.020		
B	0.77		1.65	0.030		0.065
b		0.5			0.020	
b1		0.25			0.010	
D			20			0.787
E		8.5			0.335	
e		2.54			0.100	
e3		17.78			0.700	
F			7.1			0.280
l			5.1			0.201
L		3.3			0.130	
Z			1.27			0.050



16 PACKAGE MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			1.75			0.069
a1	0.1		0.25	0.004		0.009
a2			1.6			0.063
b	0.35		0.46	0.014		0.018
b1	0.19		0.25	0.007		0.010
C		0.5			0.020	
c1	45 (typ.)					
D	9.8		10	0.386		0.394
E	5.8		6.2	0.228		0.244
e		1.27			0.050	
e3		8.89			0.350	
F	3.8		4.0	0.150		0.157
L	0.4		1.27	0.016		0.050
M			0.62			0.024
S	8 (max.)					











MILIK  
PERPUSTAKAAN  
ITN MALANG